

INDUS UNIVERSITY



REAL TIME SHOPPING MALL SAFETY AND SECURITY SYSTEM

A thesis submitted by

Muhammad Hassan Afridi 1452-2020

Muhammad Mubashir Mallick 899-2020

Muhammad Umar Hameed 1169-2020

Supervisor

Co-Supervisor

Miss Rifat

Sir Ahsan Qadeer

**THESIS SUBMITTED TOWARDS THE PARTIAL FULFILMENT OF THE
REQUIREMENT FOR THE AWARD OF BACHLOR'S DEGREE IN
SOFTWARE ENGINEERING**

**Department of Computing, FCIT
INDUS UNIVERSITY Pakistan
SPRING-2024**

CERTIFICATE

This is to certify that the work present in this thesis entitled “Real Time Shopping Mall Safety and Security System” has been carried out by Muhammad Hassan Afridi, Muhammad Umar Hameed, and Muhammad Mubashir Mallick under our supervision. The work is genuine, original and, in our opinion, suitable for submission to the Indus University for the award of degree of BS in Software Engineering.

SUPERVISOR

Miss Rifat
Lecturer
Faculty of Computing and Information Technology
Indus University Pakistan

CO-SUPERVISOR

Sir Ahsan Qadeer
Lab Engineer
Faculty of Computing and Information Technology
Indus University Pakistan



UNDERTAKING

We Muhammad Hassan Afridi, Muhammad Mubashir Mallick, a n d Muhammad Umar Hameed here by undertake that the project titled “Real Time Shopping Mall Safetyand Security System” is the work done solely by ourselves and it has not submitted anywhere for any purpose to claim any credit. The aforementioned project of the following components/deliverables.

1. Complete Functioning Project
2. Complete Thesis Report (Hard-Copy).
3. Project DVD (Source Code, Thesis Report and Manual/Guide)

Muhammad Hassan Afridi

Muhammad Mubashir Mallick

Muhammad Umar Hameed

Chairperson

FYP Head Miss Shagufta

Dean Dr. Prof Zafar Nasir

DEDICATION

I dedicate this project to Almighty Allah, my creator, my strong pillar, my source of inspiration, wisdom, knowledge and understanding. I also dedicate this work to my family who encouraged me all the way and whose encouragement has made sure that I give it all it takes to finish that which I have started. I dedicate this work and give special thanks to my teachers for being there for me throughout the entire bachelor's program. Both of you have been my best cheerleaders. We dedicate our project work to our beloved parents, who are supporting us with unconditional love in every walk of life since we are born.

ACKNOWLEDGEMENT

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully. I would like to express my special thanks of gratitude to my group members as well as our teacher who gave us this golden opportunity to do this project which also helped us doing a lot of research and we come to know about so many new things.

ABSTRACT

Everything is going smart these days, so we offered a smart market as well as a smart safety and security system in our project. This initiative promotes improved market automation. Our solution automates electronic appliances in an efficient manner, provides theft detection and fire alarm system, and uses sensor data to control room temperature, humidity, and electrical loads with a smart safety and security system. Video surveillance has been proven to be an important component of many organizations' security and safety priorities on several occasions. This project is about the construction of an Embedded Real-Time Security System that detects motion of any unknown individual in the camera and alerts market management and security departments about odd activity in the marketplaces.

TABLE OF CONTENTS

CERTIFICATE.....	I
UNDERTAKING.....	II
DEDICATION.....	III
ACKNOWLEDGEMENT.....	IV
ABSTRACT.....	V
TABLE OF CONTENTS.....	VI
LIST OF TABLES.....	IX
LIST OF FIGURES	X
ABBREVIATIONS	XI
CHAPTER 1	1
INTRODUCTION	1
1.1 MOTIVATION AND PROJECT OVERVIEW	1
1.2 BACKGROUND.....	1
1.3 PROBLEM STATEMENT.....	2
1.4 AIM AND OBJECTIVES.....	2
1.5 SCOPE OF THE PROJECT.....	2
1.6 CONTRIBUTIONS OF THE THESIS	3
1.7 STRUCTURE OF THE THESIS	3
CHAPTER 2	5
LITERATURE REVIEW	5
2.1 BACKGROUND.....	5
2.2 RELATED WORK	5
2.3 COMPARATIVE STUDY.....	8
CHAPTER 3	10
SYSTEM DESIGN.....	10
3.1 SYSTEM METHODOLOGY	10
3.1.1 SOFTWARE REQUIREMENTS	11
3.1.2 HARDWARE REQUIREMENT.....	12
3.1.3 SYSTEM CONSTRAINTS.....	14
3.1.4 EXTERNAL INTERFACE REQUIREMENTS	14
3.2 FUNCTIONAL REQUIREMENTS.....	14

Real Time Shopping Mall Safety and Security System

3.2.1 USE CASES.....	15
3.2.2 FLOW CHART DIAGRAM.....	16
3.2.3 SEQUENCE DIAGRAM.....	18
3.2.4 ER-DIAGRAM.....	21
3.3 NON-FUNCTIONAL REQUIREMENTS.....	22
3.3.1 PERFORMANCE REQUIREMENTS	22
3.3.2 SECURITY REQUIREMENTS	23
3.4 DESIGN CONSTRAINTS.....	23
3.5 SUMMARY	23
CHAPTER 4.....	24
SYSTEM IMPLEMENTATION.....	24
4.1 SYSTEM DEVELOPMENT PROCESS.....	24
4.1.1 ARDUINO MEGA.....	24
4.1.1.1 HOW ARDUINO WORK.....	25
4.1.2 ARDUINO UNO.....	25
4.1.2 DIFFERENCES BETWEEN RASPBERRY PI AND ARDUINO.....	26
4.1.3 WHY ARDUINO?.....	27
4.1.1 FIRE SYSTEM.....	28
4.1.2 FLAME SENSOR.....	28
4.1.3 TEMPERATURE SENSOR.....	29
4.1.4 CAMERA.....	29
4.1.5 THEFT DETECTION SYSTEM	30
4.1.6 PIR SENSOR.....	30
4.1.7 VOICE RECORDER.....	31
4.1.8 GSM	32
4.2.1 SYSTEM LEVEL ARCHITECTURE.....	33
Level 1 – Fire Module	33
Level 2 – Robbery detection module	33
Level 3 – Voice Assistant	34
Level 4 – Temperature Observer	34
4.3 ALGORITHMS.....	34
4.4 DATABASE DESIGN AND DEVELOPMENT	35
4.5 CODE.....	35
4.6 SUMMARY	57
CHAPTER 5	58

RESULTS AND DISSCUSSION.....	58
5.1 RESULTS.....	58
5.2.1 THEFT DETECTION MODULE WORKING.....	64
5.2.2 FIRE DETECTION MODULE.....	72
5.2.3 VOICE ASSISTANT.....	78
5.3 SUMMARY.....	83
CHAPTER 6.....	84
CONCLUSION AND FUTURE DIRECTIONS.....	84
6.1 CONCLUSION.....	84
6.2 FUTURE DIRECTIONS.....	84
REFERENCES.....	86

LIST OF TABLES

TABLE 1: COMPARISON CHART	8
---------------------------------	---

LIST OF FIGURES

FIGURE 1 OVERVIEW OF ARDUINO.....	13
FIGURE 2 ARDUINO GPIO PINS.....	13
FIGURE 3 FUNCTIONAL HIERARCHY.....	14
FIGURE 4 USE CASE S&S SYSTEM.....	15
FIGURE 5 ACTIVITY DIAGRAM OF FIRE DETECTION.....	16
FIGURE 6 ACTIVITY DIAGRAM OF THEFT DETECTION.....	17
FIGURE 7 SEQUENCE DIAGRAM OF FIRE DETECTION MODULE.....	18
FIGURE 8 SEQUENCE DIAGRAM OF THEFT DETECTION MODULE.....	19
FIGURE 9 CLASS ROLE EXAMPLE.....	19
FIGURE 10 ACTIVATION AND EXECUTION.....	20
FIGURE 11 MESSAGE EXAMPLE.....	20
FIGURE 12 LIFELINES EXAMPLE.....	20
FIGURE 13 ENTITY RELATIONSHIP DIAGRAM.....	21
FIGURE 14 ENTITIES EXAMPLE.....	21
FIGURE 15 ATTRIBUTES EXAMPLES.....	22
FIGURE 16 CARDINALITIES EXAMPLES.....	22
FIGURE 17 ARDUINO MEGA.....	25
FIGURE 18 ARDUINO UNO.....	25
FIGURE 19 ARDUINO VS RASPBERRY PI.....	26
FIGURE 20 DETECTING THE FIRE WITH FLAME SENSOR.....	28
FIGURE 21 FLAME SENSOR.....	29
FIGURE 22 DHT11 TEMPERATURE SENSOR.....	29
FIGURE 23 CAMERA.....	30
FIGURE 24 DETECTING PERSON WITH SSD.....	30
FIGURE 25 PIR MOTION SENSOR.....	31
FIGURE 26 ISD1820 RECORDER.....	31
FIGURE 27 SIM800L GSM MODULE.....	32
FIGURE 28 ARCHITECTURE DIAGRAM OF SMART MARKET, SAFETY AND SECURITY SYSTEM.....	33
FIGURE 29 SMART SHOPPING MALL SAFETY AND SECURITY SYSTEM.....	58
FIGURE 30 DESKTOP APPLICATION FOR MANAGEMENT.....	63
FIGURE 31 HOME PAGE OF RTSSS.....	64
FIGURE 32 THEFT DETECTION.....	72
FIGURE 33 FIRE DETECTION USING HSV ALGORITHM.....	77

ABBREVIATIONS

Open C. V	Computer Visions
LDR	Light dependent Resister
HSV	hue, saturation, value
PIR	Passive infrared sensor
GPIO	General Process Input/output

CHAPTER 1

INTRODUCTION

In this chapter we provide motivation and contributions of this thesis and at the end of the chapter, we present structure of the rest of the thesis.

1.1 MOTIVATION AND PROJECT OVERVIEW

The popularity of smart appliances, smart homes, and smart offices is growing every day since they are less expensive and more secure than traditional security systems. As a result of these technologies, we presented a smart market as well as a smart safety and security system to ensure market security. Many stores in marketplaces burned down due to fire, which will result in significant loss of items present in the market. However, our smart market safety technology assisted in this scenario by instantly alerting the fire department for a specific emergency situation. The major three blocks that work together to make our system workable are contained in the process of real-time shopping malls safety and security system. Sensors provide input, which is processed by a logic processing unit, and the output is provided by the main system, which check if any action required in response to the input.

1.2 BACKGROUND

Smart shopping mall safety and security system will improve mall automation, provide comfort to peoples, and reduce the time it takes to respond to an emergency scenario. Our system includes aspects that lure people to malls while also ensuring that the mall is safe from robbery and fire. In our project, we use image processing techniques and several sensors to improve the accuracy of the fire system.

1.3 PROBLEM STATEMENT

SECURITY PERSPECTIVE:

Fire and theft both are serious security and safety hazards in markets, and a recent fire incident in the UAE market damaged 125 shops, destroying millions of dollars' worth of goods.

LIFE SAVING:

In the event of a fire, failure to leave or guided to an emergency can result in the loss of human life.

1.4 AIM AND OBJECTIVES

The primary advantage is to increase safety. The key goal is to decrease the amount of time spent to act against fire and to notify the fire rescue crew as soon as possible. Security System that takes less time to notify security department and deceive thieves and criminals. The use of voice assistant technology to assist and guide consumers in the market are crucial steps

1.5 SCOPE OF THE PROJECT

In 2019, the NFPA reported \$12.4 billion in direct property damage. Every year, millions of dollars in property is burned, people are wounded, and lives are lost as a result of a fire department's late response or failure to respond as soon as a fire is noticed. Making a system that can automatically detect and report fires at an early stage in order to avoid fire loss or suppress fires as quickly as possible, and this system will ultimately help to stabilize the country's fire economy and the enormous sum of money that is tied to it.

1.6 CONTRIBUTIONS OF THE THESIS

Members	Contribution
Muhammad Umar Hameed	Project Manager, Team leader
Muhammad Mubashir Mallick	Front-end developer
Hassan Afridi	Hardware Engineer
Muhammad Umar Hameed	Documentation, Research work

1.7 STRUCTURE OF THE THESIS

The following is the thesis structure that we followed when writing this thesis.

Introduction: A thesis structure began with introduction part, which describe the project overview, also elaborate who gave the motivation to work on this project, also discuss about the benefit of our project in our community

Literature Review: This part of our project contains of the analysis, survey and research, which is describing that how much work has been done. We learned different research papers and discovered how to do it and how many advanced functions we ought to update in this project. Throughout the help of this survey, we can overcome the issues which is facing on the future

Method: The method used in our project contains all of the information required to complete this project. What are the advantages of doing this project can be clearly seen. It will benefit both the shopping mall owner and the government because it helps to decrease major emergency situations such as fire and robbery while also entertaining customers with its voice assistance feature, which can also assist in a fire emergency.

Conclusion: Our thesis concludes with the conclusion of our project. This section concluded that our study was completed. In this section, we summarized that as a result, what our project's specific goals are. In this chapter we described about our team members their role in our project, about the aims and objective of our project. The phases of our project their modules and functionality details and elaborate in next chapters.

CHAPTER 2

LITERATURE REVIEW

In this chapter we present the project's background and literature study to clarify the preceding problem statement and its working, as well as the approaches they employed to solve such challenges. We also compare the studies that have conducted on this project. There is different project linked to our project but we introduced some new features which never seen before.

2.1 BACKGROUND

Real-time Shopping Mall Safety and Security System provide comfort to humans during shopping, and reduce the time it takes to respond to an emergency situation. Our system includes aspects that lure people to malls while also ensuring that the mall is safe from robbery and fire.

2.2 RELATED WORK

A Wi-Fi-based HAS (Home Automation System) is presented in [2] to control the electrical appliances with a smartphone app. The system is based on Wi-Fi Module ESP8266 which is connected with an Arduino Mega 2560 module. The Relay board is used to control the power of electrical appliances like fans and bulbs. Other sensors placed in the models are a fan, humidity sensor, motion sensor, buzzer, and temperature sensor. A smartphone app controls the off and on operations of these sensors in a model house that is a miniature template of a house. Similarly, an Arduino 'HAS' is proposed in [3] that integrates WLAN (Wireless Local Area Network) for controlling the electrical appliances. A smartphone app is available to control the light bulb, fan, motion, and smoke sensors. Similarly, in [4] Various devices are controlled via the system like lights, and fans, etc. Besides automation, various systems for surveillance and home security have also been proposed with IoT sensors. [5] The aim of paper of Priya

H. Pande and et al are designing a home security and monitoring system. The hardware equipment's are sensors (PIR, Magnetic and LM35 sensor), microcontroller unit, relay and cell phone. When any sensor is triggered, the SMS alarm message will be sending to the home owner here. In [6] In this research paper, have used a many ultrasonic sensors for the detection of intruders. If someone pass through the sensors, transmission will be blocked and the system knows the situation. Major Voting Mechanism (MVM) is used to turn on the video camera. provided a ARM based video surveillance using embedded remote. The system sends the captured video to the user's mobile through e-mail. Similarly, in [7] the currently built prototype of the system sends alerts to the owner over voice calls using the Internet if any sort of human movement is sensed near the entrance of his house and raises an alarm optionally upon the user's discretion. The provision for sending alert messages to concerned security personnel in case of critical situation is also built into the system. [8] J.J Padmini developed a system using IoT for power utilization and conservation in smart homes. In this system, they used an image processing system to recognize human activities and designed a home automation system that can control different household devices with the help of Wi-Fi and GSM technology. In this research paper [9] author make smart home security system based on Zigbee that protects Sarawakian homeowners against flood, smoke, and intrusion. For intruder's detection they use Haar-cascade classifier to detect person. And for fire detection they use only M2 gas sensor to detect fire and make alert in any circumstances of these type. In [10] journal paper the author trained model for Intrusion and Burglary Detection using Fuzzy Technique and alert the owner by short messaging service on real-time scenario. The design approach combines the acquisition of vibration signal obtained from the burglars/intruder(s) by microcontroller-based accelerometer, intrusion detection logic built on the Fuzzy Inference System installed and configured on a cloud infrastructure with an alerting system which communicates on a real time basis to the appropriate authorities of an intrusion/burglary. In [11] proposed research paper Reuben F proposed system for smart parking system, Intrusion detection and fire detection system with using image processing technique with help of convolutional neural network (CNN) which collect number of frames as input. After getting frames from video it will be processed using image processing

techniques for feature evaluation. They extract different features from those images regardless of the events they consist. Similarly, in [12] the proposed approach defines a complete system to detect both instance and complex events when performing emergency alerting in scenarios, such as smart cities and industry, defining a comprehensive mechanism to distributively detect emergencies and to notify any requesting emergency-based application. In this system they used OPEN CV algorithm and techniques with different types of sensors to build up the accuracy rate of existing system. In [13] proposed paper designs a smart home system in which they include security system with theft detection and sensor-based fire detection with the help of different sensors. This project is based on IoT based sensors and include a google assistant device to operate the system by voice command. Voice commands control home electronic devices (lights), while fire and theft are monitored through the use of sensors and cameras. The system test shows that voice command can control lights on and off at an accuracy of 88%. Similarly [14] in this proposed paper smart home and security system was introduce which have automate the electronic appliances of house and propose IoT base security system with using different sensors to secure the house sensors including PIR motion detector sensor and camera for person detection. In [15] research paper presents a fuzzy model for early fire detection and control as symmetry's core contribution to fuzzy systems design and application in computer and engineering sciences. We utilize a fuzzy logic technique to simulate the performance of the model using MATLAB, using six parameters: temperature, humidity, flame, CO, CO₂ and O₂ vis-à-vis the Estimated Fire Intensity Prediction (**EFIP**). Results show that, using fuzzy logic, a significant improvement in fire detection is observed with an overall accuracy rate of 95.83%. In [16] research paper use IoT sensor-based technology to detect theft and fire and alert on real-time environment and Simultaneously, the buzzer alerts the nearby neighbors. The system also consists of sprayer, which sprays the chloroform liquid on the intruders and sprinkler discharges water when the effect of the fire has been detected. In [17] proposed system is used for automate the home and integrated with security system to safe home by theft. In this paper used multiple sensors like MQ2, flame sensor, smoke sensor, moisture sensor and other sensors to automate home and alert about any alarming situation.

Comparison Chart:

S.NO	Author Name		Fire safety			Theft Detection		Voice Assistance	Temperature Control
		Year	Temperature Sensor	Smoke Sensor	Image Processing	Motion Detection	Image Processing		
1	PROPOSED SYSTEM		✓	✓	✓	✓	✓	✓	✓
2	Oussama Tahan	2020	X	X	X	✓	✓	X	X
3	Akihiko Iketani	2020	X	X	X	X	✓	X	X
4	Adnan Khalil	2020	X	X	✓	X	X	X	X
5	L. Mary Gladencel	2020	X	X	X	X	X	✓	✓
6	Wen Wang	2020	X	X	✓	X	X	X	X
7	Mahmud, S.	2019	X	X	X	✓	✓	✓	X
8	Mujahid Tabassum	2020	X	✓	X	X	✓	X	✓
9	Isah Mohammed	2020	X	X	X	✓	✓	X	X
10	F. Tribhuwan	2020	X	X	✓	X	✓	X	X
11	Daniel G. Costa	2020	X	✓	✓	X	X	X	X
12	Agus Nurcahyo	2020	X	✓	X	X	✓	✓	✓
13	Emmanuel Lule	2020	X	X	X	✓	✓	X	✓
14	S. S. V. Reddy	2021	X	✓	X	✓	✓	X	X
15	S. Kumar	2022	✓	✓	X	✓	✓	✓	X
16	P K Singh	2023	X	✓	✓	X	✓	✓	X

Table 1: Comparison Chart

2.3 COMPARATIVE STUDY

There is no such project designed to ensure the safety of shopping malls. In prior study and work, they only placed single function on system, such as a theft detection system or a fire safety system. However, in our project, the GSM module will not only detect fire and theft, but will also alert the security team about the situation in the market. The system has several essential functions that will be used in an emergency situation to conquer the problem. To improve the accuracy result of fire detection, we incorporate not only sensor-based fire detection but also uses Artificial Intelligence to spot fire by using image processing, smoke and temperature sensors.

2.4 Summary

This chapter provides an overview of a project's literature review. Throughout the research, it appears that there are numerous projects in fire or theft detection, but all of them uses the same detection feature. They all uses a single feature for detection, other projects use sensors for detection, but the final result is always inaccurate, which resulting in system inefficiency. Image processing is employed to detect a fire in some projects, but no voice assistant feature was used in any of the system. However, in our project, we consolidated many characteristics onto a single platform, allowing the system to deliver reliable results. For detecting a fire with accurate results, we use various sensors or combine an image processing technology.

CHAPTER 3

SYSTEM DESIGN

In this chapter we provide motivation and contributions of this thesis and at the end of the chapter, we present structure of the rest of the thesis.

3.1 SYSTEM METHODOLOGY

These steps are essential to the system methodology.

Phase-I:

The first step is to comprehend the difficulties that people face in real life. Everyone needs a safe environment in which to live, shop, and do other things. We're focusing on mall security. After you've grasped the problem, analyze it and come up with a solution.

Phase-II:

After analyzing the problem, we moved on to the second phase, where we developed a solution and introduced three different modules for shopping mall security. We started coding for each module separately: the fire module, the theft detection module, and the voice assistance module.

Phase-III:

The third phase's challenge is to connect all three modules on a single platform, which we did with the help of our microcontroller, and then check if our hardware components are working properly and transmitting the data we need.

Phase-IV:

We check if our project is running smoothly in the final phase. Our testing team examines the project's performance, ensures that the hardware and its components

are in good working order, tests the hardware's connectivity, and generates the correct result.

3.1.1 SOFTWARE REQUIREMENTS

The following interfaces are available on the Main System of embedded software system:

Input from the camera using Python's OPEN-CV module. Speech recognition in Python with the Speech_Recognizer module. Arduino cc is a software interface that allows Arduino and sensors to communicate.

- I. Python IDLE
- II. Arduino cc software interface
- III. Arduino Shell
- IV. Windows 10

PYTHON IDLE

IDLE (Integrated Development and Learning Environment) is an integrated development environment (IDE) for Python. The Python installer for Windows contains the IDLE module by default. IDLE can be used to execute a single statement just like Python Shell and also to create, modify, and execute Python scripts. IDLE provides a fully-featured text editor to create Python script that includes features like syntax highlighting, auto completion, and smart indent. It also has a debugger with stepping and breakpoints features.

ARDUINO CC SOFTWARE INTERFACE

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices.

WINDOWS 10

Windows 10 is a Microsoft operating system for personal computers, tablets, embedded devices and internet of things devices. Windows 10 features built-in capabilities that allow corporate IT departments to use mobile device management (MDM) software to secure and control devices running the operating system. In addition, organizations can use traditional desktop management software such as Microsoft System Center Configuration Manager.

3.1.2 HARDWARE REQUIREMENT

The core mechanism of the project is linked to cameras and an Arduino microcontroller. The main system interfaces with Arduino via GPIO pins to send and receive signals from sensors. The cameras are connected to the main system separately to provide images for the fire detection and robbery module.

There are multiple sensors are use with two microcontrollers, one for each module. Arduino mega is used in fire detection module and Arduino UNO is use for theft detection module. GSM Sim800L is also used for dialing a call in any emergency situation to rescue departments and ISD1820 module is also used for make a recorded call.

ARDUINO:

The Arduino is an open-source electronics platform that allows you to create electrical creations with simple hardware and software. One thing that all Arduino boards have in common is a microcontroller. A microcontroller is essentially a little computer. You may design and construct gadgets that interact with your environment with the Arduino. The Arduino boards are essentially a controller for electronics. They may use their internal microprocessor to receive inputs (such as light on a sensor or an item near a sensor) and convert them to outputs (Drive a motor, ring an alarm, turning on an LED, display information on an LCD).

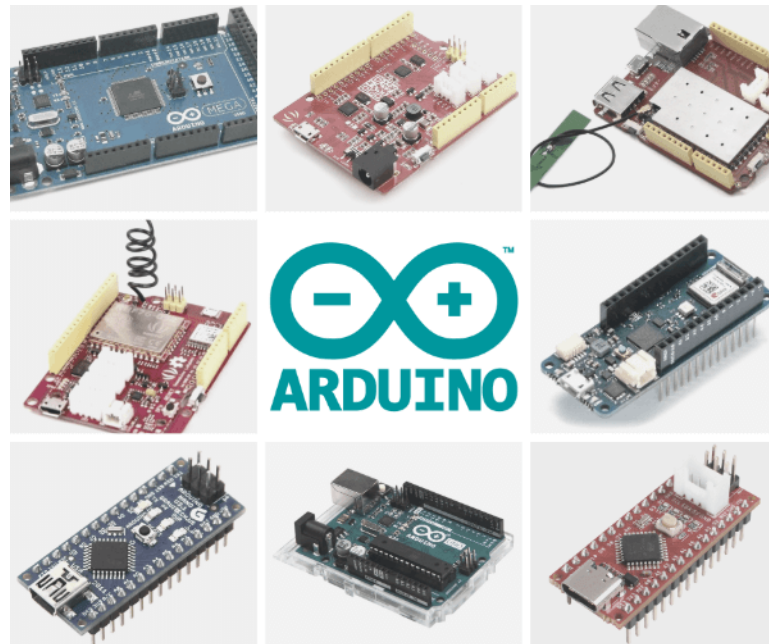


Figure 1 Overview of Arduino

GPIO PINS:

A GPIO is a signal pin on an integrated circuit or board that can be used to perform digital input or output functions. There is analog and digital both pins are available in Arduino UNO and Arduino Mega 2560. Analog pins are used to get continuous data of 0 or 1 and digital pins are used for digital inputs and outputs.

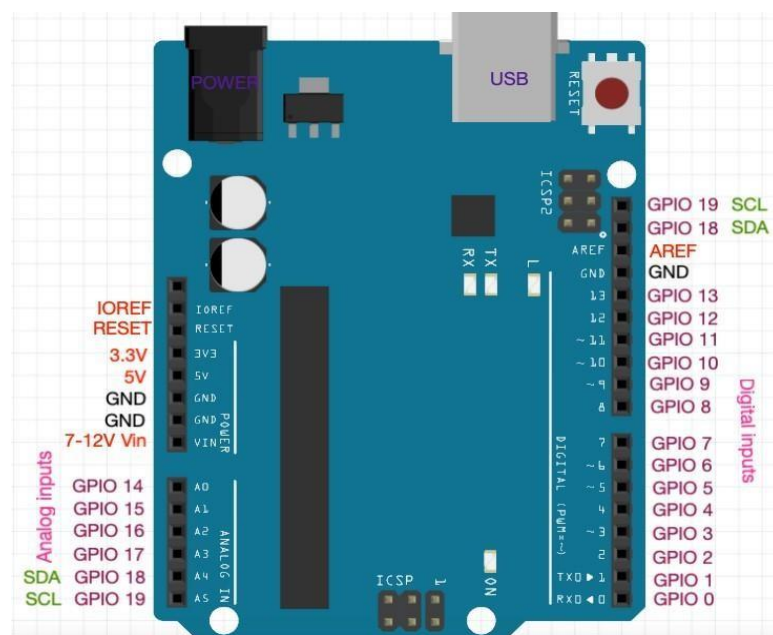


Figure 2 Arduino GPIO Pins

3.1.3 SYSTEM CONSTRAINTS

Three alternative ways for detecting fire are included in the fire detection module. To activate the GSM alert system, two requirements must be met. On the other hand, when both the motion detector sensor and the person detection are true in the robbery detection module, the GSM alerts the rescue department.

3.1.4 EXTERNAL INTERFACE REQUIREMENTS

The communication interface between the Python environment and the Arduino mega is connected with Serial library of Python, while the communication interface between the Arduino, fire, and robbery module sensors is a non-standard PWM interface using GPIO pins.

3.2 FUNCTIONAL REQUIREMENTS

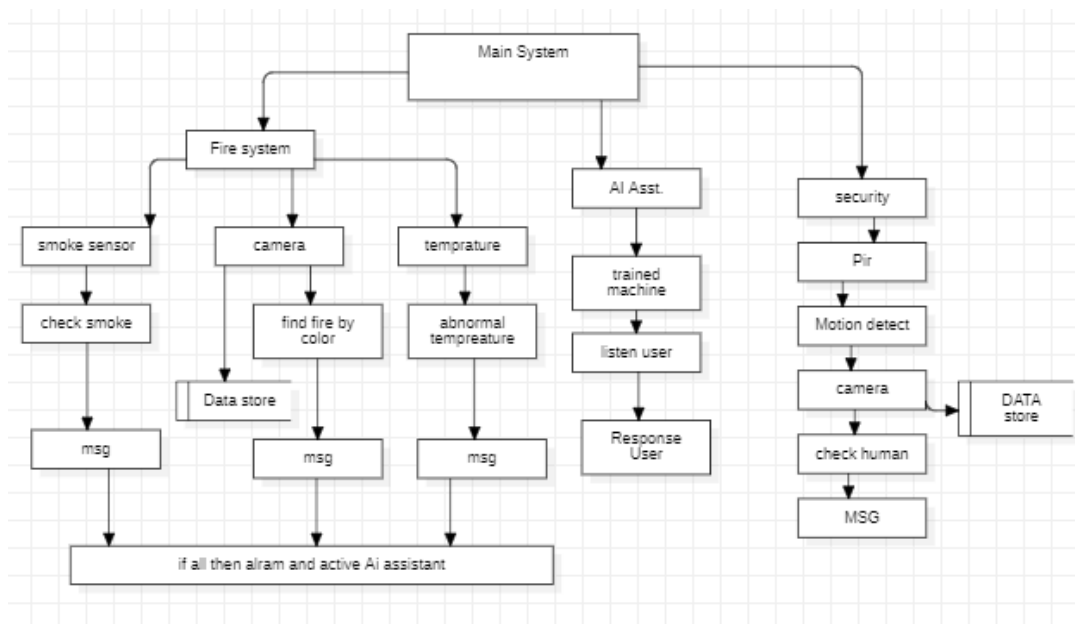


Figure 3 Functional Hierarchy

Three modules make up the real-time shopping mall safety and security system: Fire detection, Theft detection, and Voice Assistance. If there is a fire in the market, the sensor will warn the system, and the system will take some initial actions, such as alerting the rescue team and the market owner. A voice assistant module has also been built for this type of event to help the people escape from the

market. If an intruder is detected after the market has closed, the system will switch on lights and sound an alarm to deter the theft and warn the owner in real time.

3.2.1 USE CASES

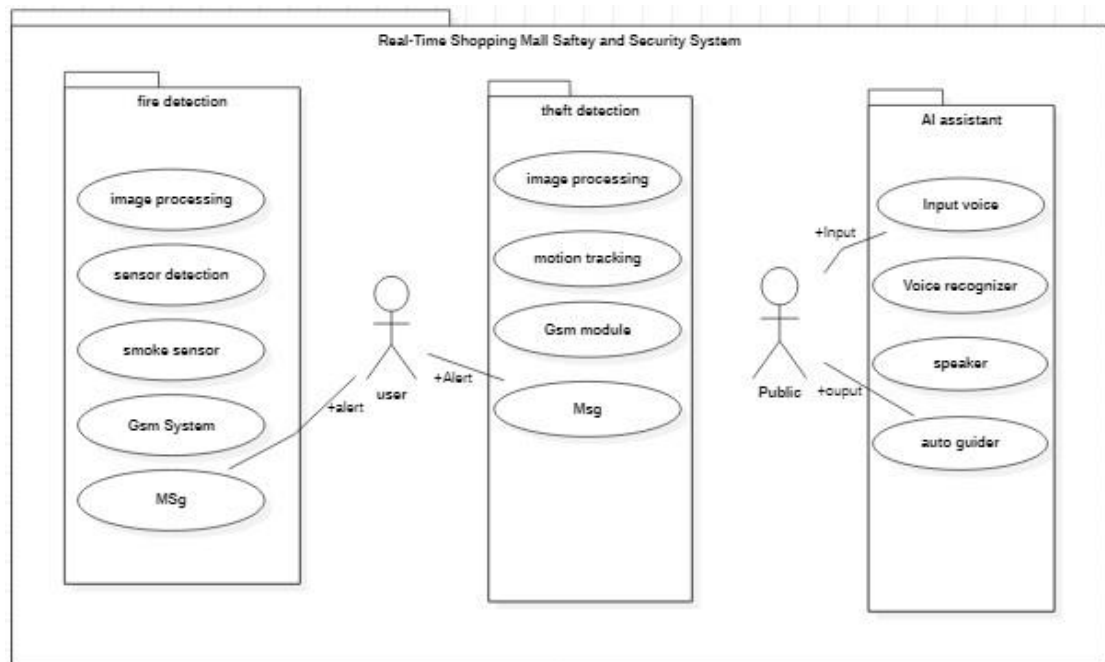


Figure 4 Use Case S&S System

The system operates on three primary functions, as shown in figure 3.4. I.e. fire detection, theft detection, and AI assistance.

In fire detection, two types of sensors are used: smoke sensor, flame sensor, and image processing technique is also included.

The second method is theft detection, which also used image processing approach as well as a motion tracking sensor.

Last but not least, there's the AI assistant, which is employed in public service. It will take the people's voice input and guide them. It may also be used to locate the fire exit in an emergency situation.

GSM system is used for both theft and fire detection.

3.2.2 FLOW CHART DIAGRAM

An activity diagram is a behavioral diagram. Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinates to represent business workflows

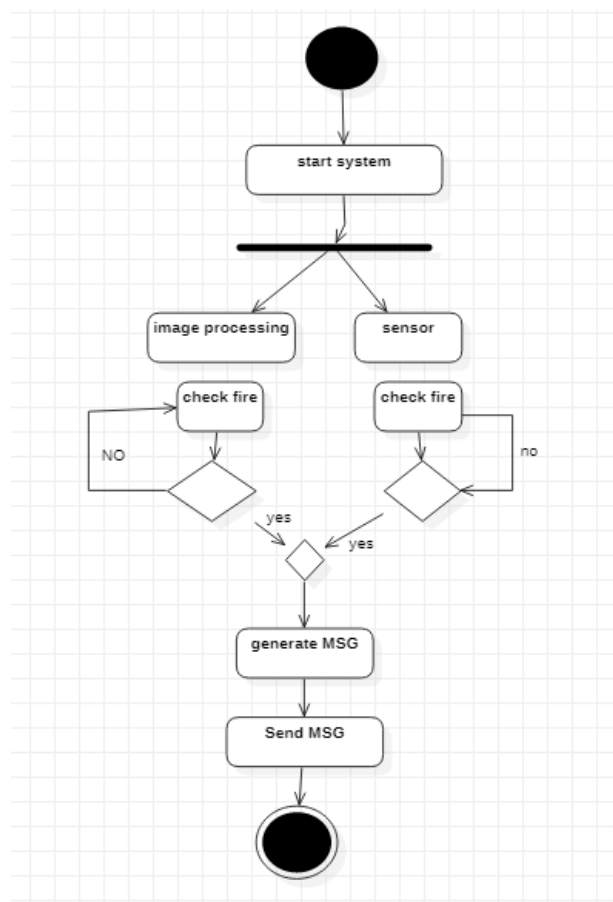


Figure 5 Activity Diagram of Fire detection

It is shown in the Fire module the flow for fire detection, which is detected by fire using image processing techniques and sensors. Furthermore, as a result of the fire, the temperature rises above the safe level, and the flames of the fire may be seen on camera and detected by sensors. The system works with all of these

activities when it checks the temperature and flame with sensors and a camera. If any abnormal condition is discovered, the system goes right and the necessary action is taken, and an alert automation system is activated, which can notify the fire department and security management team by quickly arriving on the scene.

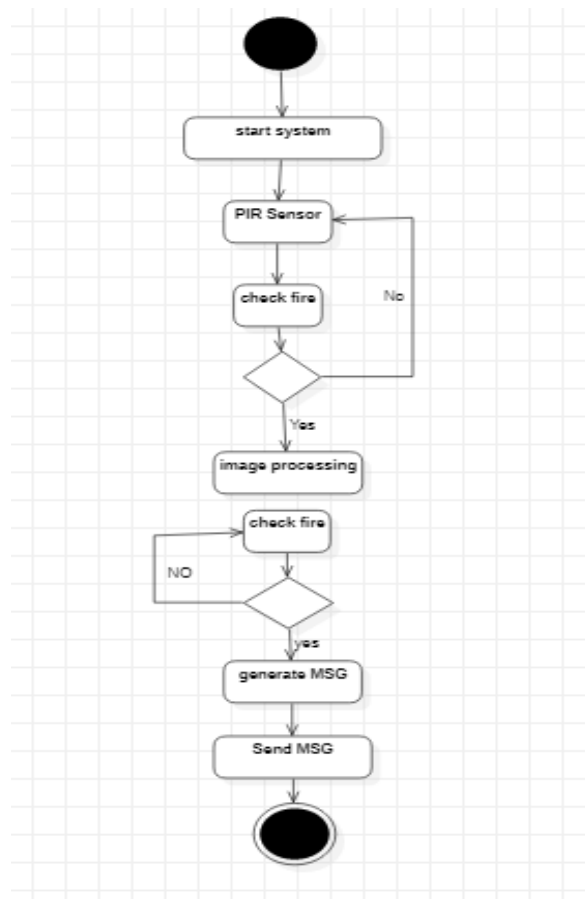


Figure 6 Activity diagram of Theft detection

The theft module explains the work flow, the detection components used in the system are PIR sensor and camera different action done such as when any movement detect on sensors then it also checks with camera which is used to monitor the activities conducted in the malls It monitors the robber's activity and sends an early warning to the mall's security management.

3.2.3 SEQUENCE DIAGRAM

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

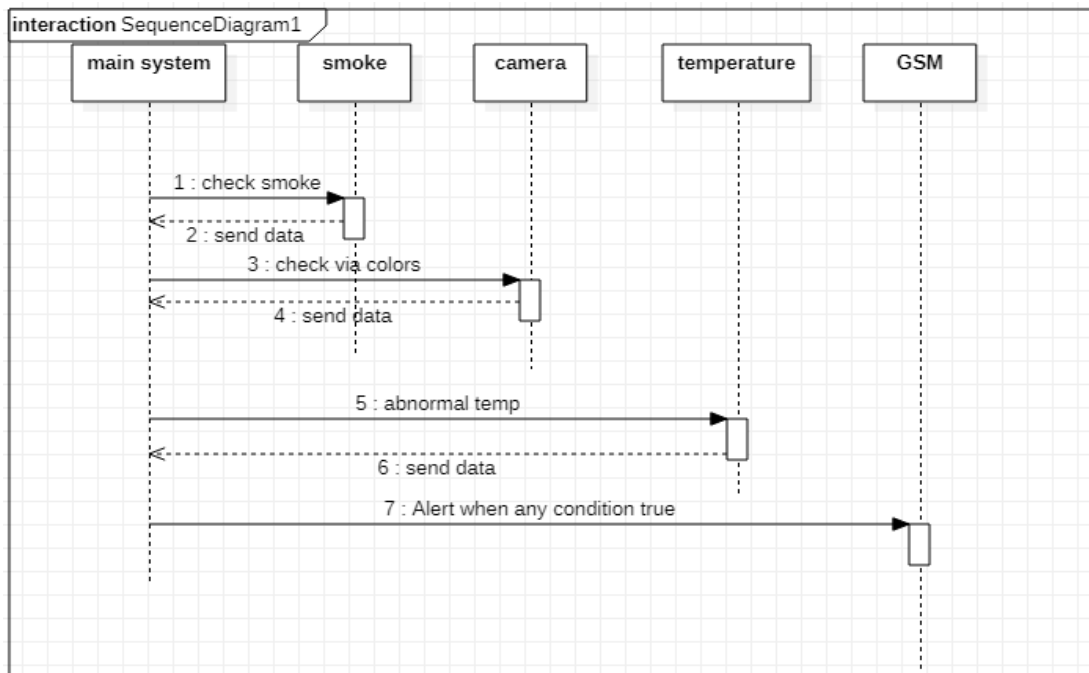


Figure 7 Sequence Diagram of Fire Detection Module

The sequence diagram for the fire detection module is shown in Figure 3.3. The main function begins with the main system. The fire detection module has two sensors: a smoke sensor and a temperature sensor. A camera is also utilized for image processing. The main system sends a message to the smoke sensor, which checks for smoke and then sends the data back to the main system. Similarly, the camera and temperature sensor get messages from the main system. If two of the conditions are met, the main system sends a message to GSM, which then sends an alarm call to emergency services.

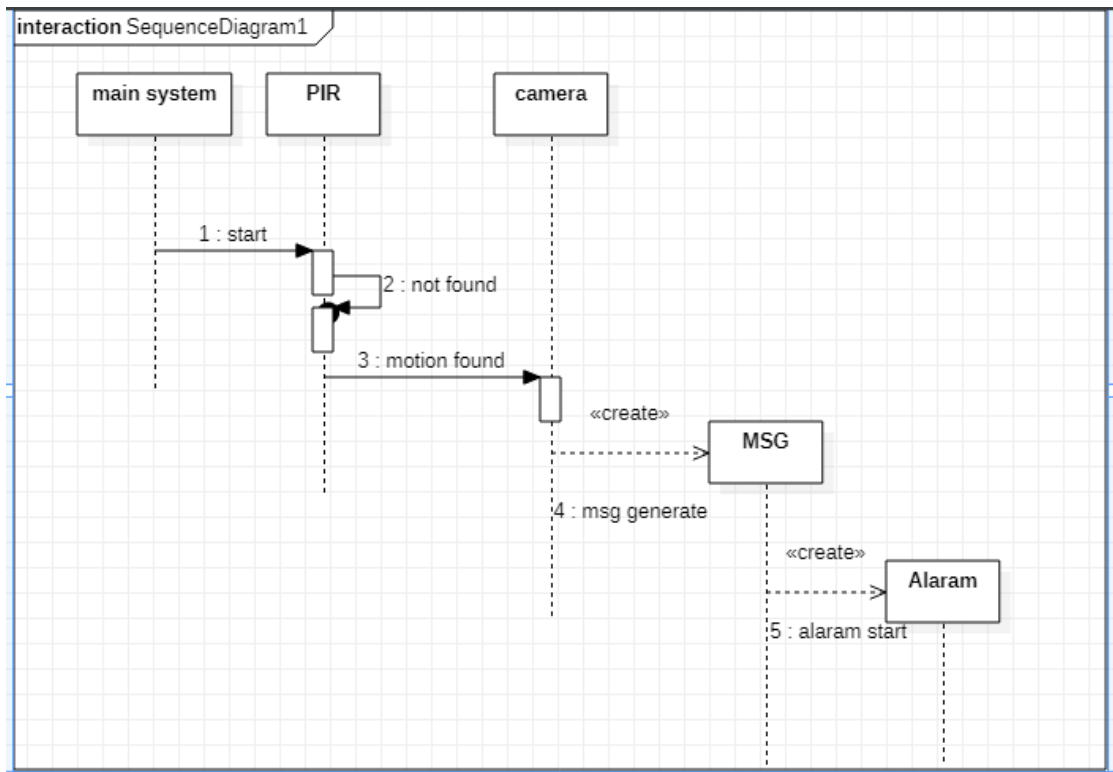


Figure 8 Sequence Diagram of theft detection module

In the theft detection module, a PIR sensor is utilized that is connected to the main system. Once the shopping mall is closed, the main system will send a message to the PIR sensor to detect any motion. If motion is detected, the main system will send an alert to the security department. If the PIR sensor detects motion and the camera detects a human, a message is sent to the security department through GSM, the lights turn on, and the alarm start in the mall.

CLASS ROLES OR PARTICIPANT:

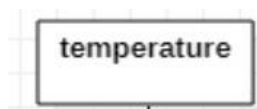


Figure 9 Class Role Example

The way an object behaves in context is described by its class role. To demonstrate class duties, use the UML object symbol, but don't specify object characteristics.

ACTIVATION OR EXECUTION OCCURRENCE:

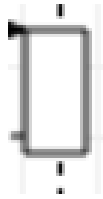


Figure 10 Activation and Execution

The time it takes for an item to accomplish a job is represented by activation boxes. Use a thin gray rectangle put vertically on an object's lifeline while it is busy running a procedure or waiting for a reply message.

MESSAGE:



Figure 11 Message Example

Messages are arrows that indicate object communication. Asynchronous messages are represented as half arrowed lines. Asynchronous messages are delivered by objects that do not wait for a response from the receiver before proceeding with their duties.

LIFELINES:



Figure 12 Lifelines Example

The presence of an object across time is shown by lifelines, which are vertical dashed lines.

3.2.4 ER-DIAGRAM

The ER, or Entity Relational Model, is a conceptual data model diagram at a high level. The Entity-Relation model is built on the concept of real-world entities and their interactions. The relationships of an entity set recorded in a database are depicted in an entity relationship diagram. To put it another way, ER diagrams assist you in explaining the logical structure of databases. An ER diagram appears to be quite similar to a flowchart at first glance. The ER Diagram, on the other hand, has numerous specific symbols, and the meanings of these symbols distinguish this model.

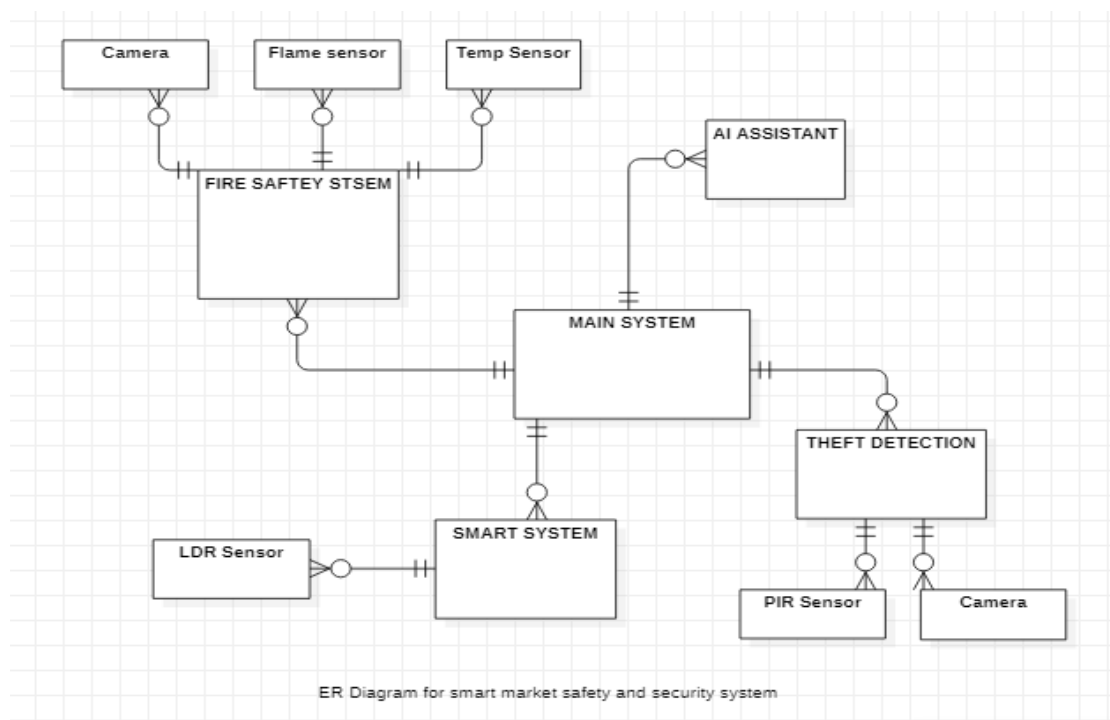


Figure 13 Entity Relationship Diagram

ENTITIES

Which are represented by rectangles. An entity is an object or concept about which you want to store information.



Figure 14 Entities Example

ATTRIBUTES

Which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity

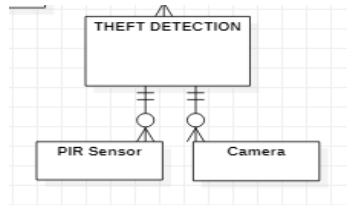


Figure 15 Attributes Examples

CARDINALITY

Specifies how many instances of an entity relate to one instance of another entity.

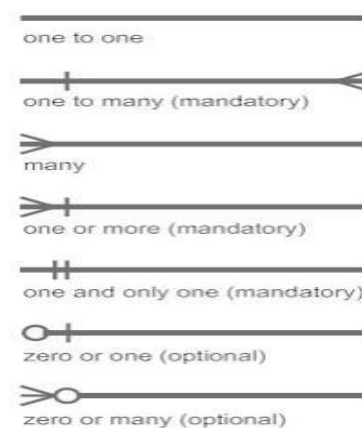


Figure 16 Cardinalities Examples

3.3 NON-FUNCTIONAL REQUIREMENTS

Following are the non-functional requirements described below:

3.3.1 PERFORMANCE REQUIREMENTS

Robustness: The security and safety system at the mall should be capable of dealing with and changing situations without compromising performance.

Quickness: When a sensor detects motion or detects a fire, it immediately alerts the microcontroller.

Failure Handling: In the case of inevitable failures, the voice assistant will provide any appropriate output rather than terminating.

Accuracy: Our system works accurately with the assistance of several sensors, taking measures in response to emerging situations and acting in real time to avert a major incident.

3.3.2 SECURITY REQUIREMENTS

Unauthorized users should not be able to take advantage of the system. If there is an electricity outage, the system will not shut down; instead, it will continue to operate and provide warnings with the help of an external battery. Using a flame or temperature sensor, our system detects a fire and then processes images using the HSV algorithm, which can respond quickly to management by sending a warning through the GSM module. It can also detect any motion via a PIR sensor at night and then perform an action for an emerging condition.

3.4 DESIGN CONSTRAINTS

A design constraint is a restriction on the needs and/or operation conditions that a robot is anticipated to perform under. A design constraint, for example, can effect the robot's shape, operation features, and usefulness. A design limitation might also be related to other factors such as manufacturing technology or the available budget for robot construction. The system must run on a portable device that also support camera. In addition, the processor speed and graphics card quality should be sufficient to run the program. The system's portability is determined by the portability of the device on which it will run. Because this is a real-time application and notebooks will be utilized to maintain speed, the system can be used everywhere the customer has his notebook.

3.5 SUMMARY

This chapter covers the overall system design. We summaries our system's design description in this part, which may describe what should be system limitations, hardware or software specifics, the system's all function and non-function requirements, and the system's real flow.

CHAPTER 4

SYSTEM IMPLEMENTATION

In this chapter we present how system is constructed. We discussed about the development process, system architecture, system installation, and system testing. Sensors are programmed with a certain algorithm. And test the system overall performance.

4.1 SYSTEM DEVELOPMENT PROCESS

Propose work based on fire detection, theft detection, voice assistant, and GSM automation alert, which is made of several modules or methods. The entire system is controlled by an Arduino Mega 2560 board, which is a microcontroller. Sensors monitor or track the surroundings, and if any unusual conditions arise, the sensor communicates the information to the microcontroller, which then generates an output. It's programmed on the Arduino software IDE. It contains 54 analogue and digital input and output pins.

4.1.1 ARDUINO MEGA

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

4.1.1.1 HOW ARDUINO WORK

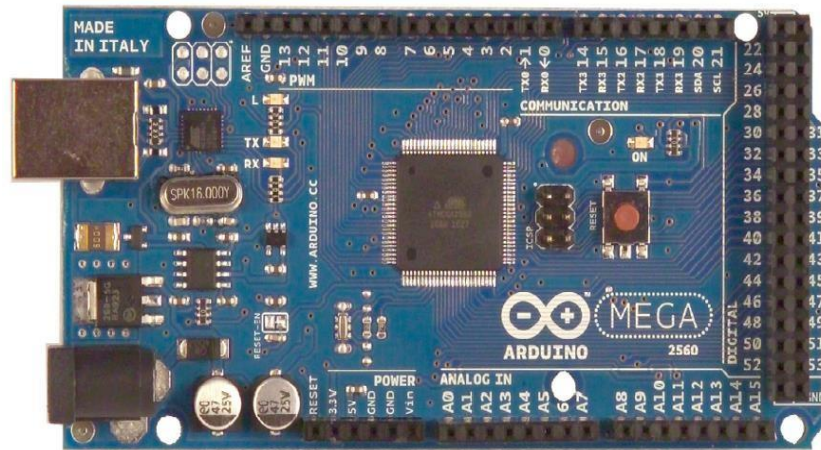


Figure 17 Arduino mega

4.1.2 ARDUINO UNO

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your Uno without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.



Figure 18 ARDUINO UNO

4.1.2 DIFFERENCES BETWEEN RASPBERRY PI AND ARDUINO

The main difference between them is: Arduino is microcontroller board, while Raspberry Pi is a microprocessor based mini-computer (SBC). The Microcontroller on the Arduino board contains the CPU, RAM and ROM. All the additional hardware on Arduino Board is for power supply, programming and IO Connectivity. Raspberry Pi SBC has all features of a computer with a processor, memory, storage, graphics driver, connectors on the board. Raspberry Pi needs an Operating System to run. Arduino doesn't need any operating system. All you need is a binary of the compiled source code. Raspberry Pi is good for developing software applications using Python, while Arduino is good for interfacing Sensors and controlling LEDs and Motors. The power requirements of Raspberry Pi and Arduino are completely different. Even though they both are powered by USB (micro-USB for Raspberry Pi and USB Type B for Arduino), Raspberry Pi needs more current than Arduino. So, you need a power adapter for Raspberry Pi but you can power Arduino from the USB port of a Computer.

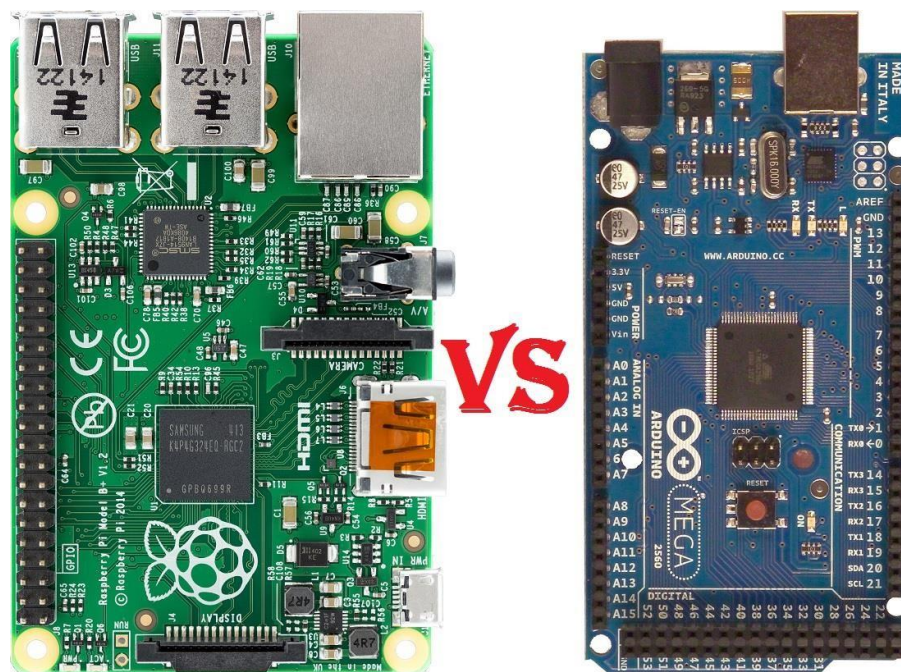


Figure 19 Arduino vs Raspberry Pi

4.1.3 WHY ARDUINO?

Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- i. **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50
- ii. **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- iii. **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- iv. **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

- v. **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

4.1.1 FIRE SYSTEM

In the fire module, fire detection is performed using image processing techniques and sensor-based detection, with the camera continuing to monitor and follow the mall's condition. Sensors can also detect changes in the surroundings utilizing flame and temperature sensors.



Figure 20 Detecting the Fire with flame sensor

4.1.2 FLAME SENSOR

A flame sensor, as the name suggests, is used for detecting and responding to the presence of a fire or flame. The flame sensor usually responds in the form of sound alarms, fuel line deactivations (such as a natural gas line or a propane line), and activation of different types of fire suppression systems. Flame sensors are commonly used to check whether the furnaces are working properly. These sensors are also employed in an ignition system to take accurate actions or to notify the operator. Flame sensors or flame detectors are comparatively more accurate and swifter than smoke or heat sensors.

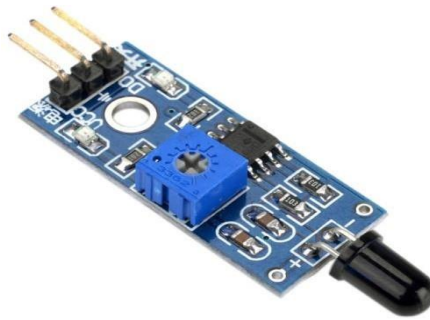


Figure 21 Flame Sensor

4.1.3 TEMPERATURE SENSOR

Temperature Sensors measure the amount of heat energy or even coldness that is generated by an object or system, allowing us to “sense” or detect any physical change to that temperature producing either an analogue or digital output.

There are many different types of Temperature Sensor available and all have different characteristics depending upon their actual application.

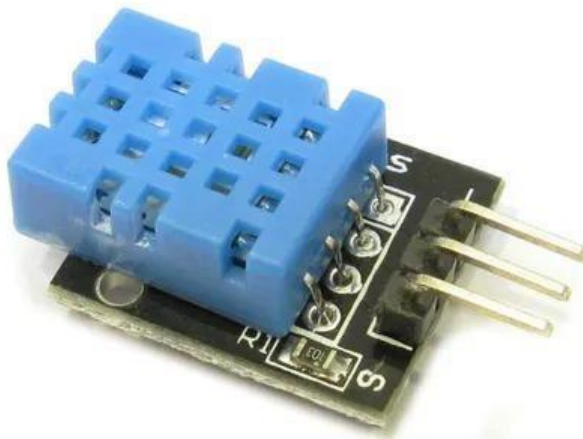


Figure 22 dht11 Temperature sensor

4.1.4 CAMERA

A device that consists of a lightproof chamber with an aperture fitted with a lens and a shutter through which the image of an object is projected onto a surface for recording (as on a photosensitive film or an electronic sensor) or for translation into electrical impulses (as for television broadcast).



Figure 23 Camera

4.1.5 THEFT DETECTION SYSTEM

The system detects theft utilizing a PIR motion sensor and image processing method in the theft module. When a PIR motion sensor detects movement, the camera uses image processing to check for human presence. If a person is discovered, the system will send a notification to the market management and security departments.



Figure 24 Detecting person with SSD

4.1.6 PIR SENSOR

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.



Figure 25 PIR Motion Sensor

4.1.7 VOICE RECORDER

ISD1820 is a small Voice Recorder and Playback module that can do the multi-segment recording. The user can achieve a high quality of recording (for 8 to 20secs) for each application with the adjustment of the on-board resistor. This Voice Recorder/Playback module is designed with embedded-Flash memory, which can hold data for up to 100 years and erase/record the life cycle up to 100,000.

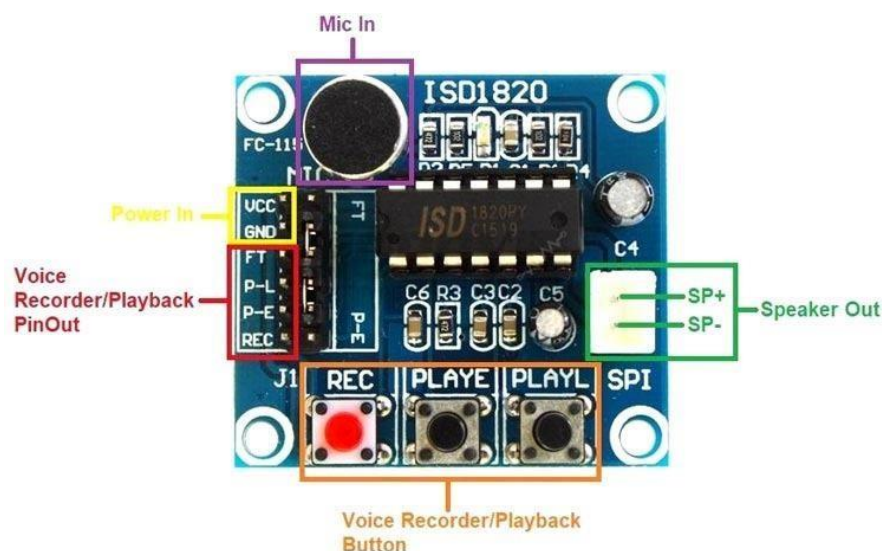


Figure 26 ISD1820 Recorder

4.1.8 GSM

The GSM stand for Global System for Mobile Communication. In our project GSM Sim800L is used for making a real time call in any emergency situation like Fire and other scenario. When this type of incident occur in shopping mall The real time safety and security system make auto recorded call to a departments which can help out the people like in fire it will call Fire Brigade in real time environment. Sim800L is most widely used GSM module it is cheaper and easy to use in environment it can communicate with microcontroller to respond in any emergency situation.



Figure 27 SIM800L GSM Module

VOICE ASSISTANT

The voice assistant responds to the user's inquiry when they have a question or ask for instructions in the mall. It's designed to be the best guider in the event of a fire, when people are unsure where to find an escape gate or which way to walk to save their lives. People may simply discover the direction of exit gate by following the instructions provided by this system. In normal situation it will guide people about any shop location cafeteria location and other locations.

4.2. SYSTEM ARCHITECTURE

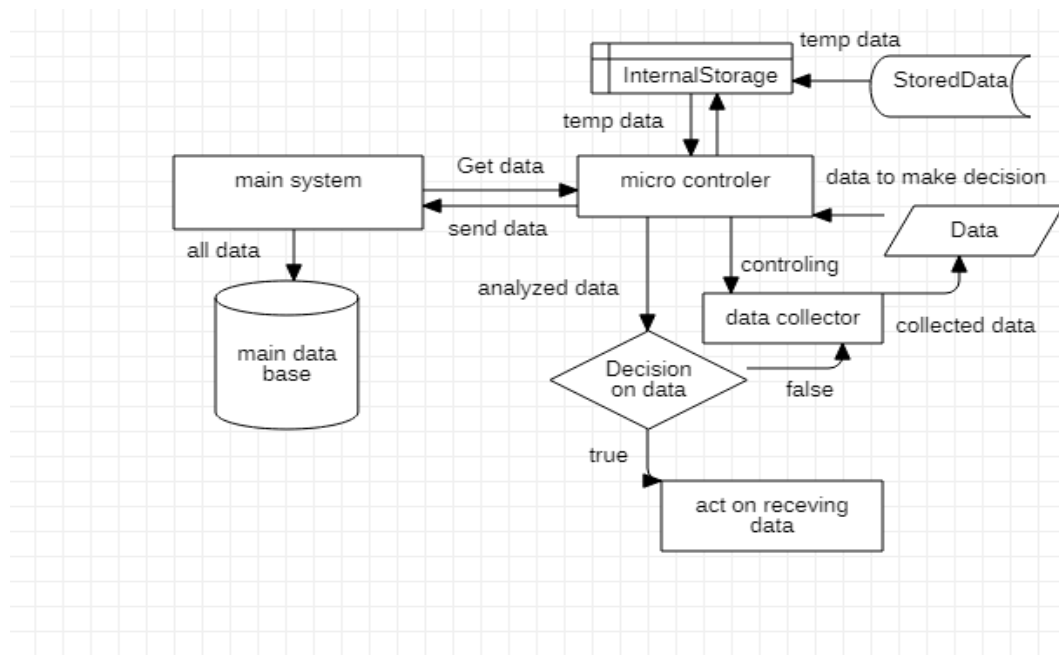


Figure 28 Architecture Diagram of Smart Market, Safety and Security system

In above figure, we have our own database, which is a temporary. Database collects the data of our micro controller. Live camera video will be saved in the database for two days, after which the film will be automatically deleted. Data is collected from the sensors and sent to the micro-controller through the data collector. If any unusual activity discovered, the micro-controller will communicate it and GSM act on receiving data.

4.2.1 SYSTEM LEVEL ARCHITECTURE

LEVEL 1 – FIRE MODULE.

The fire module was created to check for fire emergencies. It is designed to check if there are any fire emergency exits in malls, and if there are, it will automatically contact emergency services.

LEVEL 2 – ROBBERY DETECTION MODULE.

The robbery detection module was created with security in mind. It will be

activated after the mall has closed and will check if any human motion has been detected. If the message is true, it will automatically contact the security department.

LEVEL 3 – VOICE ASSISTANT.

The Voice Assistant has two functions. In the event of a fire, the first is to direct the public to any outlet, and the second is to direct the public to a fire exit.

LEVEL 4 – TEMPERATURE OBSERVER.

The temperature observer is an attached sensor to the fire module. It measures the mall's temperature. The temperature sensor will send a message to the fire module if it detects an abnormal temperature.

4.3 ALGORITHMS

The suggested system employs the SSD algorithm and the HSV method. We used the SSD algorithm to identify the theft and the HSV method to detect the fire. The fire detection using HSV models provide an accuracy of 80%.

SSD ALGORITHM

SSD stands for single shot detector. Instead of using sliding window, SSD divides the image using a grid and have each grid cell be responsible for detecting objects in that region of the image. Detection objects simply means predicting the class and location of an object within that region. SSD is designed for object detection in real-time. Faster R-CNN uses a region proposal network to create boundary boxes and utilizes those boxes to classify objects. While it is considered the state-of-the-art in accuracy, the whole process runs at 7 frames per second. Far below what real-time processing needs. SSD speeds up the process by eliminating the need for the region proposal network. To recover the drop-in accuracy, SSD applies a few improvements including multi-scale features and default boxes. These improvements allow SSD to match the Faster R-CNN's accuracy using lower resolution images, which further pushes the speed higher. According to the following comparison, it achieves the real-time processing speed and even beats

the accuracy of the Faster R-CNN.

HSV

HSV (Hue, Saturation and Value) defines a type of color space. It is similar to the modern RGB and CMYK models. The HSV color space has three components: hue, saturation and value. 'Value' is sometimes substituted with 'brightness' and then it is known as HSB. The HSV model was created by Alvy Ray Smith in 1978. HSV is also known as the hex-cone color model. The HSV color space is more efficient than RGB in estimating flame regions. The spectrum of flame colors is located in a local band in the hue spectrum. As the hue component is circular angle data.

4.4 DATABASE DESIGN AND DEVELOPMENT

The Smart Shopping Mall Safety and Security System has a storage capacity of 32 GB. The live video will be captured in the database if a theft or fire is detected the footage will be stored in database. Within 48 hours, or two days, the film will be automatically removed.

4.5 CODE

```
import tkinter

from tkinter import *

from PIL import ImageTk

from tkinter import messagebox

from tkinter import Frame

import sys

import os

class login:
```

Real Time Shopping Mall Safety and Security System

```
def __init__(self, window):

    self.window = window

    window.title("Smart Survelience System")

    #window.geometry("600x500+400+120")

    window.geometry("900x600+200+50")

    #BG img.

    self.bg=ImageTk.PhotoImage(file="image2.jpeg")

self.bg_image=Label(self.window,image=self.bg).place(x=0,y=0,relwidth=1,relh
eight=1)

    self.window.resizable(False,False)

    #Create frame

    frame=Frame(self.window,bg="white")

    frame.place(x=190,y=60,width=500,height=360)

    #create labels

    title=Label(frame,text="Admin          Login",font=("times          new
roman",25,"bold"),fg="chocolate3").place(x=180,y=30)

    sub_title=Label(frame,text="Real-time      Safety      and      Security      System
",font=("Goudy old style",15,"bold"),fg="chocolate3").place(x=130,y=90)

    #user name field

    user_name=Label(frame,text="User          name",font=("Goudy          old
style",15,"bold"),fg="grey",bg="white").place(x=130,y=140)

    user_var=StringVar()
```


Real Time Shopping Mall Safety and Security System

```
self.user_entry=Entry(frame,font=("times new roman",15),bg="lightgray")

self.user_entry.place(x=130,y=170,width=300,height=35)

#password field

user_pass=Label(frame,text="Password",font=("Goudy old
style",15,"bold"),fg="grey",bg="white").place(x=130,y=210)

pass_var=StringVar()

self.user_pass=Entry(frame,font=("times new roman",15),bg="lightgray")

self.user_pass.place(x=130,y=240,width=300,height=35)

#button
login_btn=Button(frame,text="Login",cursor="hand2",command=self.btn_action
,font=("times new roman",20),fg="white",bg="chocolate3").place(x=180,y=300,width=180,height=
40)

#message box

def btn_action(self):

    if self.user_entry.get()=="sehrishishaq" and self.user_pass.get()=="123":

        messagebox.showinfo("", "Access
successfully..!WELCOME",parent=self.window)

    def second_window(self):

        new_window=tkinter.Toplevel(window)

        #window.destroy()

        new_window.geometry("900x600+200+50")

        new_window.title("Real-Time Safety and Security System")
```

```
new_window.resizable(False,False)

#background img

self.new_bg=ImageTk.PhotoImage(file="eatoncamera.jpg")

self.newImage=Label(new_window,image=self.new_bg).place(x=0,y=0,relwidth
=1,relheight=1)

#Create frame

new_frame=Frame(new_window,bg="gray89")

new_frame.place(x=140,y=140,width=590,height=220)

#create labels

title=Label(new_frame,text="Real-Time      Safety      and      Security
System",font=("times new roman",22,"bold"),fg="chocolate3").place(x=70,y=50)

#button

fire_btn=Button(new_frame,text="Fire
Detection",command=fireBtn,cursor="hand2",font=("times new
roman",16),fg="white",bg="chocolate3").place(x=80,y=150,width=140,height=4
0)

theft_btn=Button(new_frame,text="Theft      Detec",command=
theftBtn,cursor="hand2",font=("times new
roman",16),fg="white",bg="chocolate3").place(x=240,y=150,width=140,height=
40)

voice_btn=Button(new_frame,text="Voice      Assistant",command=
voiceBtn,cursor="hand2",font=("times new
roman",16),fg="white",bg="chocolate3").place(x=400,y=150,width=140,height=
40)
```

```
        second_window(self)

    elif self.user_entry.get()==" " and self.user_pass.get()==" ":

        messagebox.showinfo("", "Blank not allowed",parent=self.window)

    else:

        messagebox.showinfo("", "invalid")

#function

def fireBtn():

    print("Fire Alert")

    os.startfile('F:/Sehrish project/Fire with frontend complete/Fire With front-
end.py')

def theftBtn():

    print("Theft Detect")

    os.startfile('F:/Sehrish project/front_end/person detection frontend.py')

def voiceBtn():

    print("Query plz")

    os.startfile('F:/Sehrish project/voice assistant/Personal_Assistant.py')

window = Tk()

obj=login(window)

window.mainloop()
```

Code of Theft Detection Module is describe below:

```
import tkinter

import cv2

import PIL.Image, PIL.ImageTk

import time

import serial

import time

#path of model file

file1path="MobileNetSSD_deploy.prototxt"

file2path="MobileNetSSD_deploy.caffemodel"

net = cv2.dnn.readNetFromCaffe(file1path, file2path)

# Labels of Network.

classNames = {

    15: 'person'

}

class App:

    def __init__(self, window, window_title, video_source=1):

        self.window = window

        self.window.title(window_title)

        self.video_source = video_source
```

Real Time Shopping Mall Safety and Security System

```
# open video source (by default this will try to open the computer webcam)

self.vid = MyVideoCapture(self.video_source)

# Create a canvas that can fit the above video source size

self.canvas = tkinter.Canvas(window, width = self.vid.width, height =
self.vid.height)

self.canvas.pack()

# Button that lets the user take a snapshot

self.btn_snapshot=tkinter.Button(window,      text="Snapshot",      width=50,
command=self.snapshot)

self.btn_snapshot.pack(anchor=tkinter.CENTER,  expand=True)

# After it is called once, the update method will be automatically called every
delay milliseconds

self.delay = 15

self.update()

self.window.mainloop()

def snapshot(self):

# Get a frame from the video source

ret, frame = self.vid.get_frame()

if ret:

    cv2.imwrite("frame-" + time.strftime("%d-%m-%Y-%H-%M-%S") +
".jpg", cv2.cvtColor(frame, cv2.COLOR_RGB2BGR))
```

```
def update(self):

# Get a frame from the video source

    ret, frame = self.vid.get_frame()

    frame_resized = cv2.resize(frame,(300,300)) # resize frame for prediction

    blob = cv2.dnn.blobFromImage(frame_resized, 0.007843, (300, 300),
(127.5, 127.5, 127.5), False)

#Set to network the input blob

    net.setInput(blob)

#Prediction of network

    detections = net.forward()

#Size of frame resize (300x300)

    cols = frame_resized.shape[1]

    rows = frame_resized.shape[0]

#For get the class and location of object detected,

# There is a fix index for class, location and confidence

# value in @detections array .

    for i in range(detections.shape[2]):

        confidence = detections[0, 0, i, 2] #Confidence of prediction

        if confidence > 0.5: # Filter prediction

            class_id = int(detections[0, 0, i, 1]) # Class label

# Object location
```

```
xLeftBottom = int(detections[0, 0, i, 3] * cols)

yLeftBottom = int(detections[0, 0, i, 4] * rows)

xRightTop    = int(detections[0, 0, i, 5] * cols)

yRightTop    = int(detections[0, 0, i, 6] * rows)

# Factor for scale to original size of frame

heightFactor = frame.shape[0]/300.0

widthFactor  = frame.shape[1]/300.0

# Scale object detection to frame

xLeftBottom = int(widthFactor * xLeftBottom)

yLeftBottom = int(heightFactor * yLeftBottom)

xRightTop    = int(widthFactor * xRightTop)

yRightTop    = int(heightFactor * yRightTop)

# Draw location of object

cv2.rectangle(frame,    (xLeftBottom,    yLeftBottom),    (xRightTop,
yRightTop),

                (0, 255, 0))

# Draw label and confidence of prediction in frame resized

if class_id in classNames:

    counter:

    count=0

    theft_alert = False
```

```
label = classNames[class_id] + ": " + str(confidence)

labelSize, baseLine = cv2.getTextSize(label,
cv2.FONT_HERSHEY_SIMPLEX, 0.5, 1)

yLeftBottom = max(yLeftBottom, labelSize[1])

cv2.rectangle(frame, (xLeftBottom, yLeftBottom - labelSize[1]),
                (xLeftBottom + labelSize[0], yLeftBottom +
baseLine),

                (255, 255, 255), cv2.FILLED)

cv2.putText(frame, label, (xLeftBottom, yLeftBottom),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0))

print(label) #print class and confidence

count=count+1

print(count)

print("Theft Alert in Camera")

##Getting 1st serial input from motion sensor and check the result

ser = serial.Serial('COM9', 9600)

b=ser.readline()

string_n = b.decode()

string = string_n.rstrip()

alert = string
```



```
print(alert)

time.sleep(0.1)

ser.close()

if alert=="Theft Alert":

    arduino=serial.Serial('COM3', 9600)

    time.sleep(2)

    arduino.write(b'1')

    print("Emergency Theft Alert")

    arduino.close()

    time.sleep(10)

if ret:

    self.photo = PIL.ImageTk.PhotoImage(image =
PIL.Image.fromarray(frame))

    self.canvas.create_image(0, 0, image = self.photo, anchor = tkinter.NW)

    self.window.after(self.delay, self.update)

class MyVideoCapture:

    def __init__(self, video_source=0):

        # Open the video source

        self.vid = cv2.VideoCapture(video_source)

        if not self.vid.isOpened():

            raise ValueError("Unable to open video source", video_source)
```

```
# Get video source width and height

self.width = self.vid.get(cv2.CAP_PROP_FRAME_WIDTH)

self.height = self.vid.get(cv2.CAP_PROP_FRAME_HEIGHT)

def get_frame(self):

    if self.vid.isOpened():

        ret, frame = self.vid.read()

        if ret:

            # Return a boolean success flag and the current frame converted to BGR

            return (ret, cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))

        else:

            return (ret, None)

    else:

        return (ret, None)

# Release the video source when the object is destroyed

def __del__(self):

    if self.vid.isOpened():

        self.vid.release()

# Create a window and pass it to the Application object

App(tkinter.Tk(), "Tkinter and OpenCV")
```

Below is the code of Fire Alert Module:

```
import tkinter

import cv2

import PIL.Image, PIL.ImageTk

import time

import numpy as np

import serial

class App:

    def __init__(self, window, window_title, video_source=1):

        self.window = window

        self.window.title(window_title)

        self.video_source = video_source

        #Open video source (by default this will try to open the computer webcam)

        self.vid = MyVideoCapture(self.video_source)

        # Create a canvas that can fit the above video source size

        self.canvas = tkinter.Canvas(window, width = self.vid.width, height =

self.vid.height)

        self.canvas.pack()

        # Button that lets the user take a snapshot

        self.btn_snapshot=tkinter.Button(window,      text="Snapshot",      width=50,

command=self.snapshot)

        self.btn_snapshot.pack(anchor=tkinter.CENTER, expand=True)
```

#After it is called once, the update method will be automatically called every delay milliseconds

```
self.delay = 15

self.update()

self.window.mainloop()

def update(self):

    fire_count = 0

    fire_alert= False

# Get a frame from the video source

    ret, frame = self.vid.get_frame()

    blur = cv2.GaussianBlur(frame,(15,15),0)

    hsv = cv2.cvtColor(blur, cv2.COLOR_BGR2HSV)

    lower = [0,74,200]

    upper = [18,166,230]

    lower = np.array(lower,dtype='uint8')

    upper = np.array(upper,dtype='uint8')

    mask = cv2.inRange(hsv,lower,upper)

    output = cv2.bitwise_and(frame,hsv,mask=mask)

    number_of_total = cv2.countNonZero(mask)

    if int(number_of_total) > 10000:

        print("Fire Alert in Camera")
```

##Getting 1st serial input from temperature sensor

```
ser = serial.Serial('COM3', 9600)
```

```
b=ser.readline()
```

```
string_n = b.decode()
```

```
string = string_n.rstrip()
```

```
temp_read = string
```

```
print(temp_read)
```

##Getting 2nd serial input from Flame sensor:

```
b = ser.readline()
```

```
string_n = b.decode()
```

```
string = string_n.rstrip()
```

```
flame_read = string
```

```
if flame_read=='FIRE ALERT':
```

```
    fire_count+1
```

```
    fire_alert= True
```

```
    print(f"{flame_read} in flame sensor")
```

```
elif flame_read=='FIRE not ALERT':
```

```
    print(f"{flame_read} in flame sensor")
```

```
time.sleep(0.1)
```

```
ser.close()
```

##Both camera and sensor alert than buzzer on:

```
if int(number_of_total) > 5000 and fire_alert==True:

    arduino=serial.Serial('COM3', 9600)

    time.sleep(2)

    arduino.write(b'1')

    print("buzzer turned ON")

    arduino.close()

if ret:

    self.photo          =          PIL.ImageTk.PhotoImage(image          =
PIL.Image.fromarray(output))

    self.canvas.create_image(0, 0, image = self.photo, anchor = tkinter.NW)

    self.window.after(self.delay, self.update)

class MyVideoCapture:

    def __init__(self, video_source=0):

# Open the video source

        self.vid = cv2.VideoCapture(video_source)

        if not self.vid.isOpened():

            raise ValueError("Unable to open video source", video_source)

# Get video source width and height

        self.width = self.vid.get(cv2.CAP_PROP_FRAME_WIDTH)

        self.height = self.vid.get(cv2.CAP_PROP_FRAME_HEIGHT)

    def get_frame(self):
```

```
        if self.vid.isOpened():

            ret, frame = self.vid.read()

            if ret:

                # Return a boolean success flag and the current frame converted to BGR

                return (ret, frame)

            else:

                return (ret, None)

        else:

            return (ret, None)

    # Release the video source when the object is destroyed

    def __del__(self):

        if self.vid.isOpened():

            self.vid.release()

    #Create a window and pass it to the Application object

    App(tkinter.Tk(), "Tkinter and OpenCV")
```

Below is the code of Voice Assistant:

```
import pyttsx3

from pyttsx3 import voice

import os

import datetime

import wikipedia
```

```
import webbrowser

import speech_recognition as sr

import pyaudio

import random

# Creating a Global Variable of Voice Assistant:

count = 0

new_rate = 150

engine = pyttsx3.init()

voices = engine.getProperty('voices')

print(voices[1].id)

engine.setProperty('voice', voices[1].id)

engine.setProperty('rate', new_rate)

def speak(audio):

    #this funtion is for computer speaking#

    engine.say(audio)

    engine.runAndWait()

def greetings():

    hour = int(datetime.datetime.now().hour)

    if hour<=12:

        speak("Hello, Good Morning Sir")

    elif hour<=17:
```



```
        speak("Hello, Good Afternoon Sir")

    elif hour<=22:

        speak("Hello, Good Evening Sir")

    elif hour>22 and hour<=24:

        speak("Hello, Good Night Sir")

    speak("I am Sera. How can I help you Sir.")

def takecommand():

    r = sr.Recognizer()

    with sr.Microphone() as source:

        print("Listening...")

        r.adjust_for_ambient_noise(source, duration = 1)

        r.pause_threshold = 1

        audio = r.listen(source)

    try:

        print("recognizing...")

        query = r.recognize_google(audio, language='en-uk')

        print(f"user said: {query}\n")

    except Exception as e:

        print("say that again please...")

        return "none"

    return query
```

#Starting a main class here:

```
if __name__ == '__main__':

    greetings()

    while True:

        if count > 0:

            speak("Anything else sir")

            main_Query=takecommand().lower()

            print(main_Query)

            if 'wikipedia' in main_Query:

                speak("searching in wikipedia..... ")

                main_Query = main_Query.replace("wikipedia", "")

                results = wikipedia.summary(main_Query, sentences=2)

                speak("According to wikipedia")

                print(results)

                speak(results)

                count=count+1

            elif 'facebook' in main_Query:

                webbrowser.open('facebook.com')

                count=count+1

            elif 'code' in main_Query:
```

```
code_path="C:\\Users\\alish\\AppData\\Local\\Programs\\Microsoft VS  
Code\\Code.exe"
```

```
os.startfile(code_path)
```

```
count=count+1
```

```
elif 'time' in main_Query:
```

```
strtime = datetime.datetime.now().strftime('%H:%M:%S')
```

```
speak(strtime)
```

```
count=count+1
```

```
elif 'cafeteria' in main_Query:
```

```
speak("it is on fourth floor")
```

```
elif 'canteen' in main_Query:
```

```
speak("it is on fourt floor")
```

```
elif 'food court' in main_Query:
```

```
speak("it is on fourt floor")
```

```
elif 'sana safinaz' in main_Query:
```

```
speak("it is on third floor ..shop number 3")
```

```
elif 'sana safinaaz' in main_Query:
```

```
speak("it is on third floor ..shop number 3")
```

```
elif 'gul ahmed' in main_Query:
```

```
speak("it is on third floor ..shop no 1")
```

```
elif 'al karam' in main_Query:
```

```
        speak("it is on second floor ..shop no 3")

elif 'jay dot' in main_Query:

    speak("it is on first floor ..shop no 3")

elif 'washroom' in main_Query:

    speak("please turn left, then go straight.")

elif 'bathroom' in main_Query:

    speak("please turn left, then go straight.")

elif 'emergency exit' in main_Query:

    speak("please turn right, then go straight, take left and their is stairs which
let you toward emergency exit gate.")

elif 'exit' in main_Query:

    speak("please turn left, then go straight, take right and their is main exit
gate.")

elif 'Fire' in main_Query:

    speak("please turn right, then go straight, take left and their is stairs which
let you toward emergency exit gate.")

elif 'none' in main_Query:

    speak("say that again please")
```

4.6 SUMMARY

The goal of this chapter is to provide a high-level design framework for us to build our project around (Real-time safety and security system). The overall system implementation is covered in this chapter. This section summarizes our system's design description, which may include system architecture design, usage of database, system mathematical model, the development phase and system's overall design. The system is controlled by an Arduino Mega 2560 board, which is a microcontroller. Sensors monitor or track the surroundings, and if any unusual conditions arise, the sensor communicates the information to the microcontroller, which then generates an output. Live camera video will be saved for two days, after which the film will be automatically deleted. Smart Shopping Mall Safety and Security System has a storage capacity of 32 GB. If a theft or fire is detected the footage will be stored in database for 48 hours, or two days, and then removed. The system uses the SSD algorithm and the HSV method.

CHAPTER 5

RESULTS AND DISSCUSSION

5.1 RESULTS

The image below is Smart Shopping Mall Safety and Security System module.

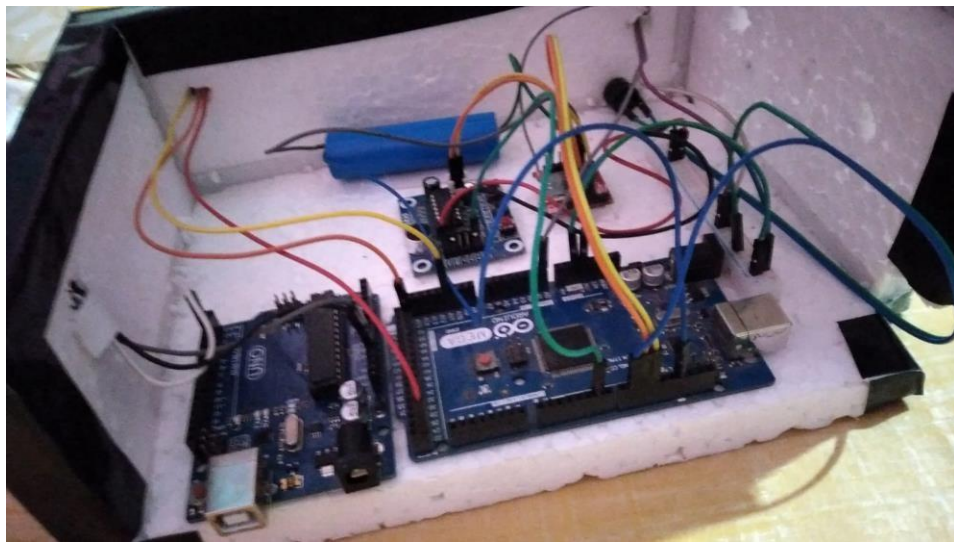


Figure 29 Smart Shopping Mall Safety and Security System

FRONT-END OF OUR DESKTOP APPLICATION:

This is the home page of our application, and a login id is necessary. This application can only be handled by management by entering the relevant password and ID.

Here is code of Grapichal User Interface of Real Time Shopping Mall Safety and Security System:

```
import tkinter

from tkinter import *

from PIL import ImageTk
```

Real Time Shopping Mall Safety and Security System

```
from tkinter import messagebox

from tkinter import Frame

import sys

import os

class login:

    def __init__(self, window):

        self.window = window

        window.title("Smart Survelience System")

        #window.geometry("600x500+400+120")

        window.geometry("900x600+200+50")

        #BG img.

        self.bg=ImageTk.PhotoImage(file="image2.jpeg")

        self.bg_image=Label(self.window,image=self.bg).place(x=0, y=0,relwidth=1,relheight=1)

        self.window.resizable(False,False)

        #Create frame

        frame=Frame(self.window,bg="white")

        frame.place(x=190,y=60,width=500,height=360)

        #create labels

        title=Label(frame,text="Admin Login",font=("times new roman",25,"bold"),fg="chocolate3").place(x=180,y=30)
```

Real Time Shopping Mall Safety and Security System

```
sub_title=Label(frame,text="Real-time Safety and Security System",font=("Goudy old style",15,"bold"),fg="chocolate3").place(x=130,y=90)
```

```
#user name field
```

```
user_name=Label(frame,text="User name",font=("Goudy old style",15,"bold"),fg="grey",bg="white").place(x=130,y=140)
```

```
user_var=StringVar()
```

```
self.user_entry=Entry(frame,font=("times new roman",15),bg="lightgray")
```

```
self.user_entry.place(x=130,y=170,width=300,height=35)
```

```
#password field
```

```
user_pass=Label(frame,text="Password",font=("Goudy old style",15,"bold"),fg="grey",bg="white").place(x=130,y=210)
```

```
pass_var=StringVar()
```

```
self.user_pass=Entry(frame,font=("times new roman",15),bg="lightgray")
```

```
self.user_pass.place(x=130,y=240,width=300,height=35)
```

```
#button
```

```
login_btn=Button(frame,text="Login",cursor="hand2",command=self.btn_action,font=("times new roman",20),fg="white",bg="chocolate3").place(x=180,y=300,width=180, height=40)
```

```
#message box
```

```
def btn_action(self):
```

```
    if self.user_entry.get()=="sehrishishaq" and self.user_pass.get()=="123":
```



```
messagebox.showinfo("", "Access
successfully..!WELCOME",parent=self.window)

def second_window(self):

    new_window=tkinter.Toplevel(window)

    #window.destroy()

    new_window.geometry("900x600+200+50")

    new_window.title("Real-Time Safety and Security System")

    new_window.resizable(False,False)

#background img

    self.new_bg=ImageTk.PhotoImage(file="eatoncamera.jpg")

    self.newImage=Label(new_window,image=self.new_bg).place(x=0,y=0,relwidth
=1,relheight=1)

#Create frame

    new_frame=Frame(new_window,bg="gray89")

    new_frame.place(x=140,y=140,width=590,height=220)

    #create labels

    title=Label(new_frame,text="Real-Time Safety and Security
System",font=("times new roman",22,"bold"),fg="chocolate3").place(x=70,y=50)

#button

    fire_btn=Button(new_frame,text="Fire
Detection",command=fireBtn,cursor="hand2",font=("times new
roman",16),fg="white",bg="chocolate3").place(x=80,y=150,width=140,height)
```

```
        theft_btn=Button(new_frame,text="Theft          Detec",command=
theftBtn,cursor="hand2",font=("times          new
roman",16),fg="white",bg="chocolate3").place(x=240,y=150,width=140,height=
40)
```

```
        voice_btn=Button(new_frame,text="Voice          Assistant",command=
voiceBtn,cursor="hand2",font=("times          new
roman",16),fg="white",bg="chocolate3").place(x=400,y=150,width=140,height=
40)
```

```
        second_window(self)
```

```
        elif self.user_entry.get()==" and self.user_pass.get()=="":
```

```
            messagebox.showinfo("", "Blank not allowed",parent=self.window)
```

```
        else:
```

```
            messagebox.showinfo("", "invalid")
```

```
#function
```

```
def fireBtn():
```

```
    print("Fire Alert")
```

```
    os.startfile('F:/Sehrish project/Fire with frontend complete/Fire With front-
end.py')
```

```
def theftBtn():
```

```
    print("Theft Detect")
```

```
    os.startfile('F:/Sehrish project/front_end/person detection frontend.py')
```

```
def voiceBtn():
```

```
    print("Query plz")
```

Real Time Shopping Mall Safety and Security System

```
os.startfile('F:/Sehrish project/voice assistant/Personal_Assistant.py')
```

```
window = Tk()
```

```
obj=login(window)
```

```
window.mainloop()
```

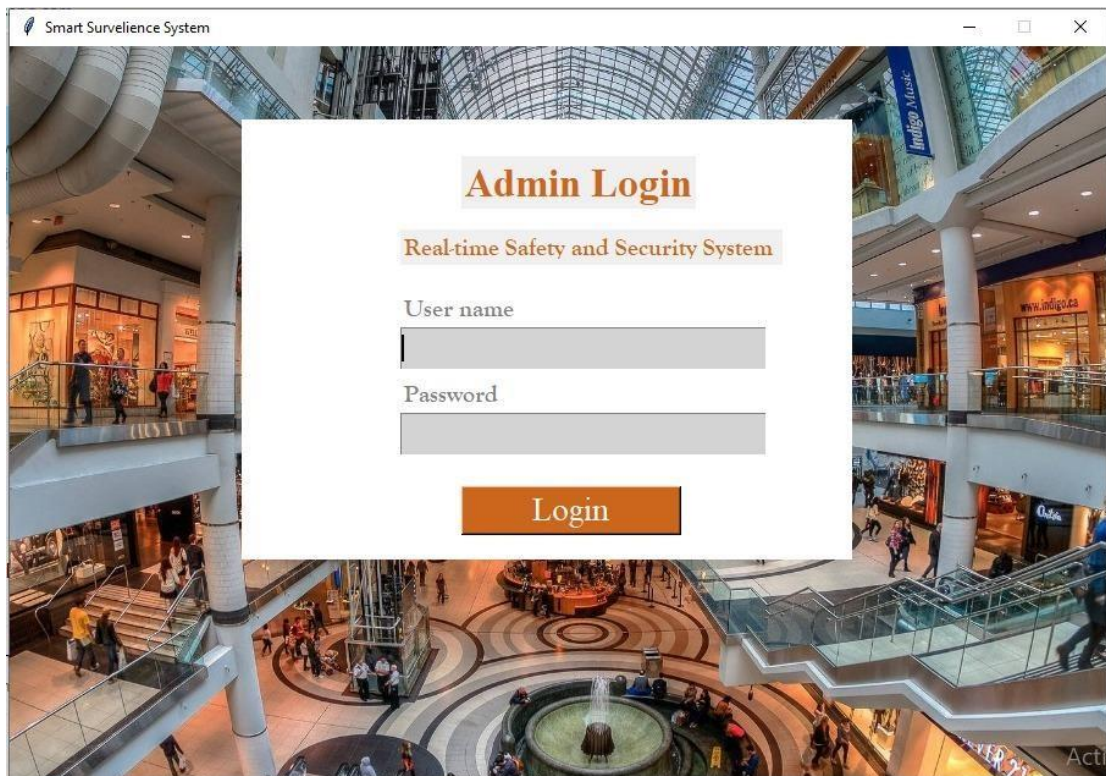


Figure 30 Desktop Application for Management

After inputting the login information the system will check the user name and password credentials to check that the person trying to login is authentic or not a not authentic person, after that it will take you toward the main window of Graphical user Interface the home page will appear, with two alternatives.

The first button which is show is for test the theft detection, and the second button is used to test the fire detection module and the third button is used for voice assistant. You can run one module at a time and can operate multiple modules at a time.

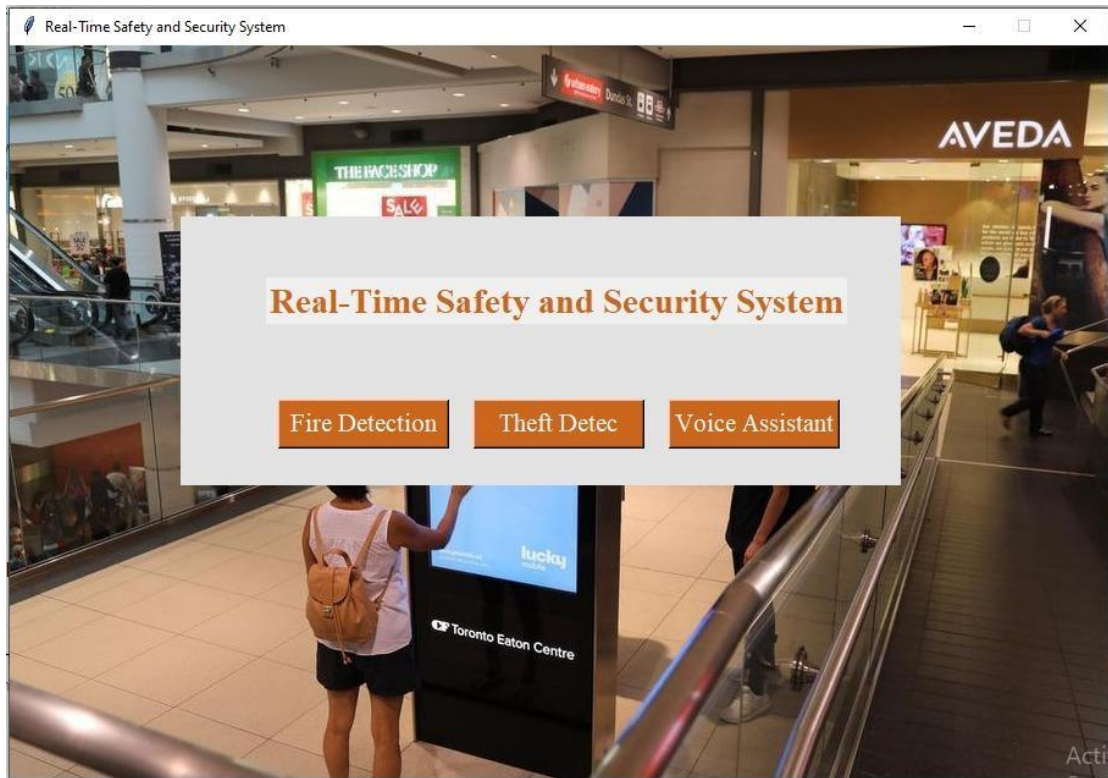


Figure 31 Home Page of RTSSS

The first button which is show is for test the theft detection, and the second button is used to test the fire detection module and the third button is used for voice assistant. You can run one module at a time and can operate multiple modules at a time.

5.2.1 THEFT DETECTION MODULE WORKING

The PIR sensor and camera in the theft detection module will continuously check for motion. If any human motion is detected in the PIR sensor and camera, an alarm message will be generated, and information will be transmitted to gsm. It is important to notice that both the PIR sensor and the camera conditions must be true; if either condition is false, no message will be sent to the gsm. If a theft is detected, the GSM will automatically initiate a call to the security department. The call contains a recorded message. The alarm buzzer will sound automatically, and all of the mall's lights will turn on.

Code of Theft Detection Module is describe below:

```
import tkinter

import cv2

import PIL.Image, PIL.ImageTk

import time

import serial

import time

#path of model file

file1path="MobileNetSSD_deploy.prototxt"

file2path="MobileNetSSD_deploy.caffemodel"

net = cv2.dnn.readNetFromCaffe(file1path, file2path)

# Labels of Network.

classNames = {

    15: 'person'

}

class App:

    def __init__(self, window, window_title, video_source=1):

        self.window = window

        self.window.title(window_title)

        self.video_source = video_source
```

```
# open video source (by default this will try to open the computer webcam)

self.vid = MyVideoCapture(self.video_source)

# Create a canvas that can fit the above video source size

self.canvas = tkinter.Canvas(window, width = self.vid.width, height =
self.vid.height)

self.canvas.pack()

# Button that lets the user take a snapshot

self.btn_snapshot=tkinter.Button(window,      text="Snapshot",      width=50,
command=self.snapshot)

self.btn_snapshot.pack(anchor=tkinter.CENTER, expand=True)

# After it is called once, the update method will be automatically called every
delay milliseconds

self.delay = 15

self.update()

self.window.mainloop()

def snapshot(self):

# Get a frame from the video source

ret, frame = self.vid.get_frame()

if ret:
```

```
cv2.imwrite("frame-" + time.strftime("%d-%m-%Y-%H-%M-%S") +
".jpg", cv2.cvtColor(frame, cv2.COLOR_RGB2BGR))

def update(self):

# Get a frame from the video source

ret, frame = self.vid.get_frame()

frame_resized = cv2.resize(frame,(300,300)) # resize frame for prediction

blob = cv2.dnn.blobFromImage(frame_resized, 0.007843, (300, 300),
(127.5, 127.5, 127.5), False)

#Set to network the input blob

net.setInput(blob)

#Prediction of network

detections = net.forward()

#Size of frame resize (300x300)

cols = frame_resized.shape[1]

rows = frame_resized.shape[0]

#For get the class and location of object detected,

# There is a fix index for class, location and confidence

# value in @detections array .

for i in range(detections.shape[2]):

confidence = detections[0, 0, i, 2] #Confidence of prediction

if confidence > 0.5: # Filter prediction
```

```
class_id = int(detections[0, 0, i, 1]) # Class label

# Object location

xLeftBottom = int(detections[0, 0, i, 3] * cols)

yLeftBottom = int(detections[0, 0, i, 4] * rows)

xRightTop    = int(detections[0, 0, i, 5] * cols)

yRightTop    = int(detections[0, 0, i, 6] * rows)

# Factor for scale to original size of frame

heightFactor = frame.shape[0]/300.0

widthFactor  = frame.shape[1]/300.0

# Scale object detection to frame

xLeftBottom = int(widthFactor * xLeftBottom)

yLeftBottom = int(heightFactor * yLeftBottom)

xRightTop    = int(widthFactor * xRightTop)

yRightTop    = int(heightFactor * yRightTop)

# Draw location of object

cv2.rectangle(frame,    (xLeftBottom,    yLeftBottom),    (xRightTop,
yRightTop),

               (0, 255, 0))

# Draw label and confidence of prediction in frame resized

if class_id in classNames:

    counter:
```



```
count=0

theft_alert = False

label = classNames[class_id] + ": " + str(confidence)

labelSize, baseLine = cv2.getTextSize(label,
cv2.FONT_HERSHEY_SIMPLEX, 0.5, 1)

yLeftBottom = max(yLeftBottom, labelSize[1])

cv2.rectangle(frame, (xLeftBottom, yLeftBottom - labelSize[1]),
               (xLeftBottom + labelSize[0], yLeftBottom +
baseLine),

               (255, 255, 255), cv2.FILLED)

cv2.putText(frame, label, (xLeftBottom, yLeftBottom),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0))

print(label) #print class and confidence

count=count+1

print(count)

print("Theft Alert in Camera")

##Getting 1st serial input from motion sensor and check the result

ser = serial.Serial('COM9', 9600)

b=ser.readline()

string_n = b.decode()
```

```
string = string_n.rstrip()

alert = string

print(alert)

time.sleep(0.1)

ser.close()

if alert=="Theft Alert":

    arduino=serial.Serial('COM3', 9600)

    time.sleep(2)

    arduino.write(b'1')

    print("Emergency Theft Alert")

    arduino.close()

    time.sleep(10)

if ret:

    self.photo = PIL.ImageTk.PhotoImage(image =
PIL.Image.fromarray(frame))

    self.canvas.create_image(0, 0, image = self.photo, anchor = tkinter.NW)

    self.window.after(self.delay, self.update)

class MyVideoCapture:

    def __init__(self, video_source=0):

        # Open the video source

        self.vid = cv2.VideoCapture(video_source)
```

```
        if not self.vid.isOpened():

            raise ValueError("Unable to open video source", video_source)

# Get video source width and height

self.width = self.vid.get(cv2.CAP_PROP_FRAME_WIDTH)

self.height = self.vid.get(cv2.CAP_PROP_FRAME_HEIGHT)

def get_frame(self):

    if self.vid.isOpened():

        ret, frame = self.vid.read()

        if ret:

# Return a boolean success flag and the current frame converted to BGR

            return (ret, cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))

        else:

            return (ret, None)

    else:

        return (ret, None)

# Release the video source when the object is destroyed

def __del__(self):

    if self.vid.isOpened():

        self.vid.release()

# Create a window and pass it to the Application object

App(tkinter.Tk(), "Tkinter and OpenCV")
```



Figure 32 Theft Detection

When a person come in front of camera the and detected by system when it is detected the system confirm from pir motion sensor that any motion occur in the environment. If both condition are true then it will raise alarm and make a call to emergency rescue department.

5.2.2 FIRE DETECTION MODULE

The fire detection module is based on two sensors and one image processing technique. The sensors are a flame and temperature sensor, and an image processing camera is employed. For the emergency call to be sent, all three circumstances must be met, namely that both the sensor and the camera identify the fire. If any of the condition is untrue, no call will be generated and If the sensors detect a fire, GSM will automatically contact the security department. The call includes a recorded message.

Below is the code of Fire Alert Module:

```
import tkinter

import cv2

import PIL.Image, PIL.ImageTk

import time

import numpy as np

import serial

class App:

    def __init__(self, window, window_title, video_source=1):

        self.window = window

        self.window.title(window_title)

        self.video_source = video_source

        #Open video source (by default this will try to open the computer webcam)

        self.vid = MyVideoCapture(self.video_source)

        # Create a canvas that can fit the above video source size

        self.canvas = tkinter.Canvas(window, width = self.vid.width, height =

self.vid.height)

        self.canvas.pack()

        # Button that lets the user take a snapshot

        self.btn_snapshot=tkinter.Button(window,      text="Snapshot",      width=50,

command=self.snapshot)
```

```
self.btn_snapshot.pack(anchor=tkinter.CENTER, expand=True)
```

#After it is called once, the update method will be automatically called every delay milliseconds

```
self.delay = 15
```

```
self.update()
```

```
self.window.mainloop()
```

```
def update(self):
```

```
    fire_count = 0
```

```
    fire_alert= False
```

```
# Get a frame from the video source
```

```
ret, frame = self.vid.get_frame()
```

```
blur = cv2.GaussianBlur(frame,(15,15),0)
```

```
hsv = cv2.cvtColor(blur, cv2.COLOR_BGR2HSV)
```

```
lower = [0,74,200]
```

```
upper = [18,166,230]
```

```
lower = np.array(lower,dtype='uint8')
```

```
upper = np.array(upper,dtype='uint8')
```

```
mask = cv2.inRange(hsv,lower,upper)
```

```
output = cv2.bitwise_and(frame,hsv,mask=mask)
```

```
number_of_total = cv2.countNonZero(mask)
```

```
if int(number_of_total) > 10000:
```

```
print("Fire Alert in Camera")

##Getting 1st serial input from temperature sensor

ser = serial.Serial('COM3', 9600)

b=ser.readline()

string_n = b.decode()

string = string_n.rstrip()

temp_read = string

print(temp_read)

##Getting 2nd serial input from Flame sensor:

b = ser.readline()

string_n = b.decode()

string = string_n.rstrip()

flame_read = string

if flame_read=='FIRE ALERT':

    fire_count+1

    fire_alert= True

    print(f"{flame_read} in flame sensor")

elif flame_read=='FIRE not ALERT':

    print(f"{flame_read} in flame sensor")

time.sleep(0.1)

ser.close()
```

##Both camera and sensor alert than buzzer on:

```
if int(number_of_total) > 5000 and fire_alert==True:
```

```
    arduino=serial.Serial('COM3', 9600)
```

```
    time.sleep(2)
```

```
    arduino.write(b'1')
```

```
    print("buzzer turned ON")
```

```
    arduino.close()
```

```
if ret:
```

```
    self.photo = PIL.ImageTk.PhotoImage(image =  
PIL.Image.fromarray(output))
```

```
    self.canvas.create_image(0, 0, image = self.photo, anchor = tkinter.NW)
```

```
    self.window.after(self.delay, self.update)
```

```
class MyVideoCapture:
```

```
    def __init__(self, video_source=0):
```

```
# Open the video source
```

```
    self.vid = cv2.VideoCapture(video_source)
```

```
    if not self.vid.isOpened():
```

```
        raise ValueError("Unable to open video source", video_source)
```

```
# Get video source width and height
```

```
    self.width = self.vid.get(cv2.CAP_PROP_FRAME_WIDTH)
```

```
    self.height = self.vid.get(cv2.CAP_PROP_FRAME_HEIGHT)
```



```
def get_frame(self):  
  
    if self.vid.isOpened():  
  
        ret, frame = self.vid.read()  
  
        if ret:  
  
# Release the video source when the object is destroyed  
  
def __del__(self):  
  
    if self.vid.isOpened():  
  
        self.vid.release()  
  
#Create a window and pass it to the Application object  
  
App(tkinter.Tk(), "Tkinter and OpenCV")
```

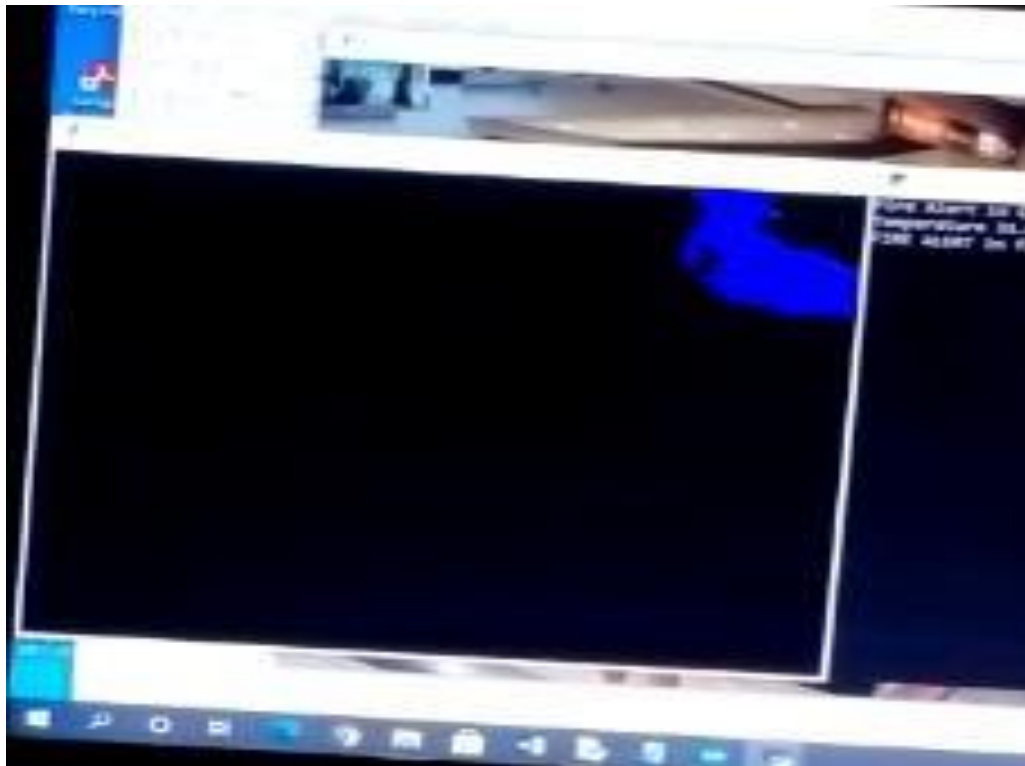


Figure 33 Fire Detection using HSV Algorithm

As shown in fig 27. When fire detected in camera the system will check through flame sensor and temperature and if they all alert about fire then it will make emergency call to the rescue department about incident of fire.

5.2.3 VOICE ASSISTANT

A voice assistant is used to direct customers to the location of a store. It can also be employed in the event of a fire. In an emergency, it will direct you to the emergency exit.

Below is the code of Voice Assistant:

```
import pyttsx3

from pyttsx3 import voice

import os

import datetime

import wikipedia

import webbrowser

import speech_recognition as sr

import pyaudio

import random

# Creating a Global Variable of Voice Assistant:

count = 0

new_rate = 150

engine = pyttsx3.init()

voices = engine.getProperty('voices')
```

```
print(voices[1].id)

engine.setProperty('voice', voices[1].id)

engine.setProperty('rate', new_rate)

def speak(audio):

    #this funtion is for computer speaking#

    engine.say(audio)

    engine.runAndWait()

def greetings():

    hour = int(datetime.datetime.now().hour)

    if hour<=12:

        speak("Hello, Good Morning Sir")

    elif hour<=17:

        speak("Hello, Good Afternoon Sir")

    elif hour<=22:

        speak("Hello, Good Evening Sir")

    elif hour>22 and hour<=24:

        speak("HELLO, Good Night Sir")

    speak("I am Sera. How can I help you Sir.")

def takecommand():

    r = sr.Recognizer()

    with sr.Microphone() as source:
```

```
    print("Listening...")

    r.adjust_for_ambient_noise(source, duration = 1)

    r.pause_threshold = 1

    audio = r.listen(source)

    try:

        print("recognizing...")

        query = r.recognize_google(audio, language='en-uk')

        print(f"user said: {query}\n")

    except Exception as e:

        print("say that again please...")

        return "none"

    return query

#Starting a main class here:

if __name__ == '__main__':

    greetings()

    while True:

        if count > 0:

            speak("Anything else sir")

        main_Query=takecommand().lower()

        print(main_Query)

        if 'wikipedia' in main_Query:
```

```
    speak("searching in wikipedia..... ")

    main_Query = main_Query.replace("wikipedia", "")

    results = wikipedia.summary(main_Query, sentences=2)

    speak("According to wikipedia")

    print(results)

    speak(results)

    count=count+1

elif 'facebook' in main_Query:

    webbrowser.open('facebook.com')

    count=count+1

elif 'code' in main_Query:

    code_path="C:\\Users\\alish\\AppData\\Local\\Programs\\Microsoft      VS
Code\\Code.exe"

    os.startfile(code_path)

    count=count+1

elif 'time' in main_Query:

    strtime = datetime.datetime.now().strftime('%H:%M:%S')

    speak(strtime)

    count=count+1

elif 'cafeteria' in main_Query:

    speak("it is on fourth floor")
```

elif 'canteen' in main_Query:

 speak("it is on fourt floor")

elif 'food court' in main_Query:

speak("it is on fourt floor")

elif 'sana safinaz' in main_Query:

 speak("it is on third floor ..shop number 3")

elif 'sana safinaaz' in main_Query:

 speak("it is on third floor ..shop number 3")

elif 'gul ahmed' in main_Query:

 speak("it is on third floor ..shop no 1")

elif 'al karam' in main_Query:

 speak("it is on second floor ..shop no 3")

elif 'jay dot' in main_Query:

 speak("it is on first floor ..shop no 3")

elif 'washroom' in main_Query:

 speak("please turn left, then go straight.")

elif 'bathroom' in main_Query:

 speak("please turn left, then go straight.")

elif 'emergency exit' in main_Query:

```
        speak("please turn right, then go straight, take left and their is stairs which  
let you toward emergency exit gate.")
```

```
    elif 'exit' in main_Query:
```

```
        speak("please turn left, then go straight, take right and their is main exit  
gate.")
```

```
    elif 'Fire' in main_Query:
```

```
        speak("please turn right, then go straight, take left and their is stairs which  
let you toward emergency exit gate.")
```

```
    elif 'none' in main_Query:
```

```
        speak("say that again please")
```

5.3 SUMMARY

In this chapter, we discussed the ultimate result of our project, the appearance (module), and the type of testing we conducted to ensure that our project was functioning properly. Smart Shopping Mall Safety and Security System do manual and exploratory testing to identify project flaws and hide them with logic solution. We talked about our project's GUI and how the theft and fire detection systems work.

CHAPTER 6

CONCLUSION AND FUTURE DIRECTIONS

In this chapter, we examine the project's conclusion and future direction, which is based on IOT, as well as the title "Real time shopping mall safety and security system" is also discussed. This chapter will wrap up the entire project or possible progress on the proposed project.

6.1 CONCLUSION

In this project, we offer a real-time solution to market security and safety concerns, as well as an introduced fire alarm system and security system to assure market security. As we have seen in the past, numerous stores in the market have been stolen, and thieves have easily left with millions of rupees. In this case, the smart market, safety, and security system would detect movements and send an alarm message to the union of market and security agencies. We didn't obtain fire information promptly in certain cases, and as a result, there were a lot of casualties and economic loss.

6.2 FUTURE DIRECTIONS

We can add some new features in the future, such as face mask detection because it is necessary to wear a mask in covid-19 situation, and a feature to count people at the entrance gate to see how many people enter the market, and the same at the exit point to manage the maximum number of people at a time during covid-19 situation.

Real time shopping mall safety and security system have multiple modules and have various future system extensions or enhancement like:

In the future extension of this project as seen a covid-19 situation and SOP's face mask detection feature can also be include to ensure on the entrance of shopping mall that all the people are wearing face mask if anyone seen without mask it will not give him or her permission to enter in shopping mall.

One more enhancement can be included in this project that install a person counter to ensure to allowed the limited persons to enter in a shopping mall at a time.

In fire module for image processing technique there should be lot of work to do to get more accurate and potential result from the system. And we can install multiple auto fire extinguisher to start spray on fire when camera notify about fire according to its axis.

In robbery detection module our system only use ultrasonic sensor and PIR sensor but we can implement more sensor to enhance more security areas like we can also implement gesture detection and object detection algorithm to extend and improved more in security of shopping malls.

There should be a lot of work and future enhancement available in voice assistant system Brands such as Amazon, Google are continuing to fuel this trend as they compete for market share. Voice interfaces are advancing at an exponential rate in industries of all kinds, ranging from healthcare to banking, as companies are racing to release their own voice technology integrations to keep pace with consumer demand.

Forty-one percent of voice assistant users are concerned about trust and privacy according to a [report](#) from Microsoft. With news from Google I/O and Amazon's MARS conferences announcing that assistants will essentially be able to plan an entire evening, for example, find local movie times, buy tickets, book a restaurant reservation and schedule an Uber, concerns regarding payments and sensitive information are valid. Voice payments, in particular, will become more secure and convenient for users to make purchases. Speaker verification and ID will also become paramount as part of the voice assistant experience with more security being built around the user.

REFERENCES

- [1] Khan Muhammad, Joel J. P. C. Rodrigues, Sergey Kozlov, Francesco Piccialli, Victor Hugo C. de Albuquerque, Energy-Efficient Monitoring of Fire Scenes for Intelligent Networks, IEEE Network, year 2020.
- [2] Akihiko Iketani*, Atsushi Nagai*, Yoshinori Kuno*y and Yoshiaki Shirai, A Novel Design for a Comprehensive Smart Automation System for the Office, Department of Computer-Controlled Mechanical Systems, Osaka University, 2020
- [3] Adnan Khalil, Sami Ur Rahman, Fakhre Alam and Irshad Khalil, A novel design for a comprehensive smart automation system for the office Department of Computer Science and Information Technology, University of Engineering and Technology, Peshawar, Pakistan, 2020.
- [4] Houache Nouredine, kechar bouabdellah, Design and implementation of a smart home system with two levels of security based on IoT technology International Journal of Distributed Systems and Technologies, Volume 11, Indonesia, Issue 3 July-September 2020.
- [5] L. Mary Gladence¹ · V. Maria Anu¹ · R. Rathna² · E. Brumancia¹, Smart Reconnaissance and Apprise System Enhanced with IoT, Volume 8, Germany, International Journal for Research in Applied Science & Engineering Technology (IJRASET), 2020
- [6] Mahmud, S., Ahmed, S., & Shikder, K., A Smart Home Automation and Metering System using Internet of Things (IoT). International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST), Beijing, 2019.
- [7] Grace Roadney Anthony, Theft Detection System, United State patent application publication grace, year 2020.
- [8] M Narayan Murth, P.AjaySaiKiran, A Smart Office Automation System, IEEE International Conference, Coimbatore, India, 2018.

- [9] Chua Boon Liang, Mujahid Tabassum, Saad Bin Abul Kashem, Smart Home Security System Based on Zigbee, Beijing: IEEE, 2020.
- [10] Isah Mohammed M., Charles Ikerionwu, Re-engineering Real-time Intrusion and Burglary Detection using Fuzzy Technique, International Journal of Computer Applications, Volume 176, April 2020.
- [11] S. L. Bangare^{1*}, Reuben F. Tribhuwan², Intelligent Video Surveillance System Based On Machine Learning, journal of engineering science, Vol 11, Amsterdam, 2020.
- [12] Daniel G. Costa , Francisco Vasques , Paulo Portugal and Ana Aguiar, On The Use of Cameras for The Detection of Critical Events in Sensors-Based Emergency Alerting Systems, Journal of sensor and actuator networks, Published: 10 October 2020.
- [13] Heru Susanto, Agus Nurcahyo, Design and Implementation of a Smart Home Security System Using Voice Command and Internet of Things, KHAZANAH INFORMATIKA Vol. 6 No. 2, Yogyakarta, October 2020.
- [14] Malti Bansal, Navodit Adarsh, Nitin Kumar, Monika Meena, 24 × 7 Smart IoT Based Integrated Home Security System, Proceedings of the Fourth International Conference on Inventive Systems and Control (ICISC), IEEE, 2020.
- [15] Emmanuel Lule, Chomora Mikeka, Alexander Ngenzi, Design of an IoT-Based Fuzzy Approximation Prediction Model for Early Fire Detection to Aid Public Safety and Control in the Local Urban Markets, Symmetry journal, August 2020.
- [16] Mr. P. Baskaran¹, Aasim Ahmed S R², Abbas Yusuf³, Deeptanshu Bordoloi, Smart Reconnaissance and Apprise System Enhanced with IoT, Amsterdam: Elsevier, 2020.
- [17] Mohammed Murad, Oguz Bayat, Hamzah M. Marhoon, Design and implementation of a smart home system with two levels of security based on IoT technology, Article in Indonesian Journal of Electrical Engineering and Computer Science, Indonesia, January 2020